

# Interactive Visualization for Computational Linguistics Tutorial at ESSLI-09

---

*Gerald Penn*

*University of Toronto, Department of Computer Science*

*Sheelagh Carpendale*

*University of Calgary, Department of Computer Science*

*Christopher Collins*

*University of Toronto, Department of Computer Science*

## Tutorial Instructors

**Gerald Penn** (gpenn@cs.utoronto.ca)

**Associate Professor, University of Toronto Computer Science**

Gerald Penn's research interests are in computational linguistics, theoretical computer science, programming languages, spoken language processing, and human-computer interaction. He is probably best known as the co-designer and maintainer of the ALE programming language, and has published widely on topics pertaining to logics and discrete algorithms for natural language processing applications. He is a member of the advisory board to [Computational Linguistics](#) and the editorial board of [Linguistics & Philosophy](#), and is a past president of the ACL Mathematics of Language Society.

**Sheelagh Carpendale** (sheelagh@ucalgary.ca)

**Associate Professor, University of Calgary Computer Science**

**Canada Research Chair: Information Visualization**

Sheelagh Carpendale holds a Canada Research Chair in Information Visualization and an NSERC/SMART/iCORE Industrial Research Chair in Interactive Technologies at the University of Calgary.

She is the recipient of several major awards including the British Academy of Film and Television Arts Award (BAFTA) for Off-line Learning, and has been involved with successful technology transfer to Idelix Software Inc. Her research focuses on the visualization, exploration and manipulation of information.

Current research includes: visualizing uncertainty particularly in medical data, visualizing biological data, developing visualizations to support computational linguistic research and the development of methodologies to support collaborative data analysis with visualization. Sheelagh Carpendale's research in information visualization and interaction design draws on her dual background in Computer Science (Ph.D. Simon Fraser University) and Visual Arts (Sheridan College, School of Design and Emily Carr, College of Art).

### Assisted by:

**Christopher Collins** (ccollins@cs.utoronto.ca)

**PhD Candidate, University of Toronto Computer Science**

Christopher Collins received his M.Sc. in the area of Computational Linguistics from University of Toronto in 2004. His PhD research focus is inter-disciplinary, combining computational linguistics and information visualization. He is currently in his final year of PhD studies, investigating interactive visualizations of linguistic data with a focus on convergence and coordination of multiple views of data to provide enhanced insight. He has developed various methods for generating, reading, and comparing visual summaries of document thematic content for everyday users and data analysts. Recent publications include a new method for revealing relationships amongst visualizations, and a system for exposing the uncertainty in statistical natural language systems. He recently embarked on a study of visualization use in a team of machine translation researchers and plans to continue collaboration with language engineers to provide them with an enhanced ability to analyse and improve their algorithms.

## Abstract

Interactive information visualization is an emerging and powerful research technique that can be used to understand models of language and their abstract representations. Much of what computational linguists fall back upon to improve NLP applications and to model language "understanding" is structure that has, at best, only an indirect attestation in observable data. An important part of our research progress thus depends on our ability to fully investigate, explain, and explore these structures, both empirically and relative to accepted linguistic theory. The sheer complexity of these abstract structures, and the observable patterns on which they are based, usually limits their accessibility — often even to the researchers creating or attempting to learn them.

To aid in this understanding, visual "externalizations" are used for presentation and explanation — traditional statistical graphs and custom-designed illustrations fill the pages of ACL papers. These visualizations provide *post hoc* insight into the representations and algorithms designed by researchers, but visualization can also assist in the process of research itself.

There are special statistical methods, falling under the rubric of "exploratory data analysis," and visualization techniques just for this purpose, in fact, but these are not widely used or even known in CL. These techniques offer the potential for revealing structure and detail in data, before anyone else has noticed them.

When observing natural language engineers at work, we also notice that, even without a formal visualization background, they often create sketches to aid in their understanding and communication of complex structures. These are *ad hoc* visualizations, but they, too, can be extended by taking advantage of current information visualization research.

This tutorial will enable members of the ESSLI community to leverage information visualization theory into exploratory data analysis, algorithm design, and data presentation techniques for their own research. We draw on fundamental studies in cognitive psychology to introduce "visual variables" — visual dimensions on which data can be encoded. We also discuss the use of interaction and animation to enhance the usability and usefulness of visualizations.

Topics covered in this tutorial include a review of information visualization techniques that are applicable to CL, pointers to existing visualization tools and programming toolkits, and new directions in visualizing CL data and results. We also discuss the challenges of evaluating visualizations, noting differences from the evaluation methods traditionally used in CL, and discuss some heuristic approaches and techniques used for measuring insight. Information visualizations in CL research can also be measured by the impact they have on algorithm and data structure design.

Information visualization is also filled with opportunities to make more creative visualizations that benefit from the CL community's deeper collective understanding of natural language. Given that most visualizations of language are created by researchers with little or no linguistic expertise, we'll cover some open and very ripe possibilities for improving the state of the art in text-based visualizations.

## **Tutorial Objectives:**

This tutorial will equip participants with:

- An understanding of the importance and applicability of information visualization techniques to computational linguistics research;
- Knowledge of the basic principles of information visualization theory;
- The ability to identify appropriate visualization software and techniques that are available for immediate use and for prototyping;
- A working knowledge of research to date in the area of linguistic information visualization.

## **Part I: Readings**

### **Theory-related Readings:**

[1] Chapter 1, Readings in Information Visualization: Using Vision to Think. Stuart Card, Jock Mackinlay, and Ben Shneiderman, Morgan Kaufmann 1999.

[2] The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations Ben Shneiderman, Proc. 1996 IEEE Visual Languages, also Maryland HCIL TR 96-13.

[3] S. Carpendale. Considering Visual Variables as a Basis for Information Visualisation. Research report 2001-693-16, Department of Computer Science, University of Calgary, Calgary, AB, Canada, 2003.

[4] Lisa Tweedie, Robert Spence, H. Dawkes, and H. and Su, "Externalising Abstract Mathematical Models," Proceedings of CHI '96, Denver CO, April 1996, pp.406-412.

[5] Ji Soo Yi, Youn ah Kang, John T. Stasko and Julie A. Jacko, "Toward a Deeper Understanding of the Role of Interaction in Information Visualization", IEEE Transactions on Visualization and Computer Graphics, (Paper presented at InfoVis '07), Vol. 13, No. 6, November/December 2007, pp. 1224-1231.

[6] S. Carpendale. (2008). Evaluating Information Visualizations. In A. Kerren, J.T. Stasko, J-D. Fekete, C. North (Eds.), Information Visualization – Human-Centered Issues and Perspectives. Vol. 4950 of LNCS State-of-the-Art Survey, pp. 19-45, Springer.

### **Applications of Visualization (in no particular order)**

[1] Fernanda B. Viégas, Martin Wattenberg, Frank van Ham, Jesse Kriss, and Matt McKeon. Many Eyes: A Site for Visualization at Internet Scale. Proc of Infovis, 2007.

[2] Christopher Collins and Sheelagh Carpendale. VisLink: Revealing Relationships Amongst Visualizations. IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization), 13(6), November--December, 2007.

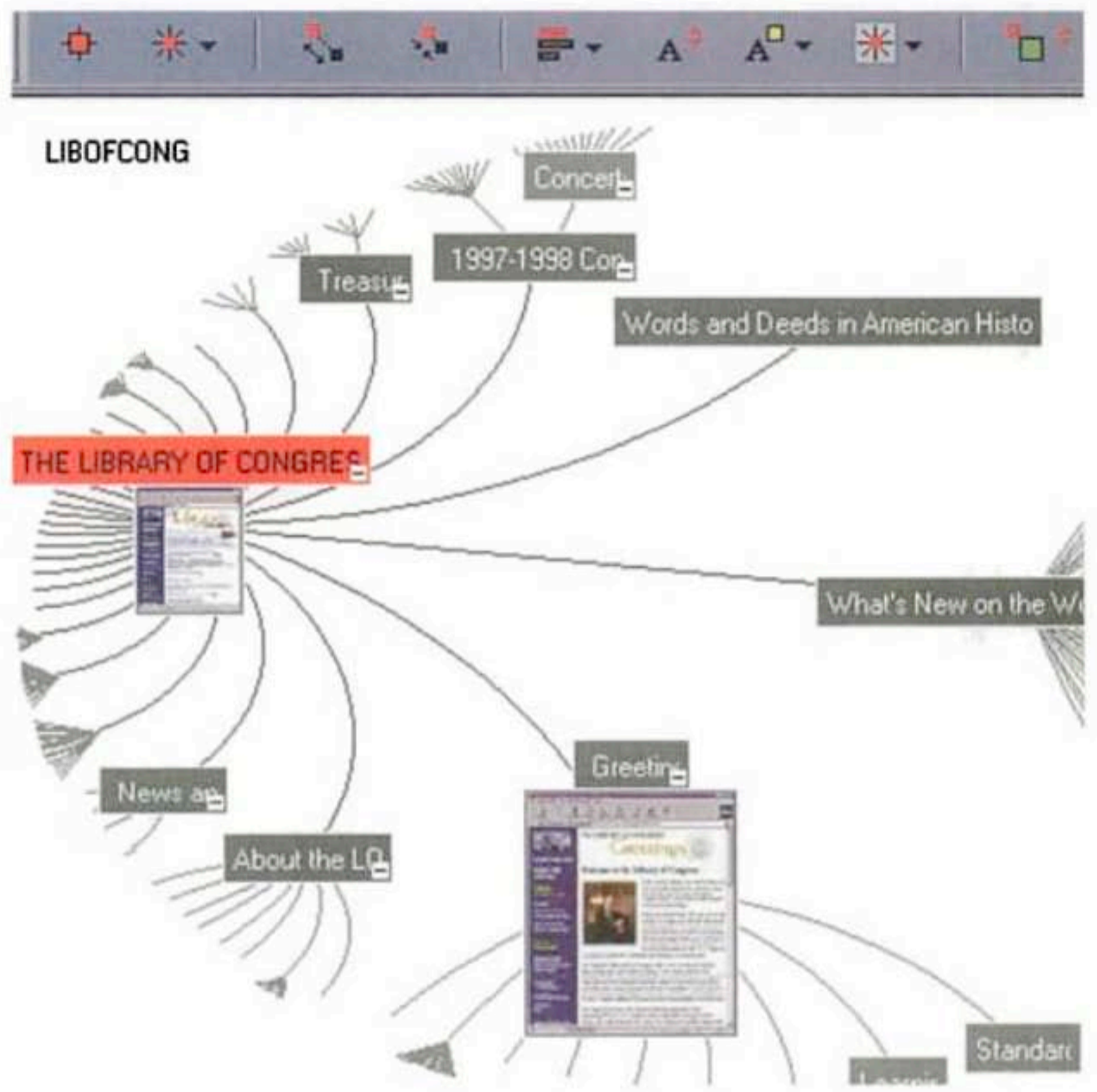
- [3] Herman, M.S. Marshall, and G. Melançon (2000) Graph Visualisation in Information Visualisation: A Survey, IEEE Transactions on Visualization and Computer Graphics, vol. 6(1), pp. 24-44.
- [4] James A. Wise and James J. Thomas and Kelly Pennock and David Lantrip and Marc Pottier and Anne Schur and Vern Crow. Visualizing the Non-Visual: Spatial Analysis and Interaction with Information from Text Documents. Proc. IEEE Symp. Information Visualization, InfoVis, pp. 51-58, IEEE Computer Soc. Press, 30-31, October 1995.
- [5] Marti A. Hearst: TileBars: Visualization of Term Distribution Information in Full Text Information Access. CHI 1995: 59-66
- [6] Brad Paley. TextArc: An alternate way to view a text. <http://www.textarc.org/> and in InfoVis'02.
- [7] Martin Wattenberg, Fernanda B. Viégas: The Word Tree, an Interactive Visual Concordance. IEEE Trans. Vis. Comput. Graph. 14(6): 1221-1228 (2008)
- [9] Martin Wattenberg. Arc Diagrams: Visualizing Structure in Strings. In Proceedings of the 2002 IEEE Symposium on Information Visualization, IEEE Computer Science Press 2002.
- [8] Petra Neumann, Annie Tat, Torre Zuk and Sheelagh Carpendale. KeyStrokes: Personalizing Typed Text with Visualization. In Proceedings of Eurographics\IEEE VGTC Symposium on Visualization, 2007.
- [10] Anthony Don, Elena Zheleva, Machon Gergory, Sureyya Tarkan, Loretta Auvil, Tanya Clement, Ben Shneiderman, and Catherine Plaisant. Discovering interesting usage patterns in text collections: Integrating text mining with visualization. In Proc. of the 2007 Conf. on Information and Knowledge Management.
- [11] Fernanda B. Viégas, Scott Golder and Judith Donath. Visualizing Email Content: Portraying Relationships from Conversational Histories. In Proc. of CHI, 2006.
- [12] Michelle L. Gregory, Nancy Chinchor, Paul Whitney, Richard Carter, Elizabeth Hetzler, and Alan Turner. 2006. User-directed sentiment analysis: Visualizing the affective content of documents. In Proc. of the Workshop on Sentiment and Subjectivity in Text, pages 23–30. ACL.
- [13] Christopher Collins, Sheelagh Carpendale, and Gerald Penn. DocuBurst: Visualizing Document Content using Language Structure. In Proc. of the Eurographics/IEEE-VGTC Symposium on Visualization (EuroVis), 2009.
- [14] Joshua Albrecht, Rebecca Hwa, and G. Elisabeta Marai. The Chinese Room: Visualization and Interaction to Understand and Correct Ambiguous Machine Translation. In Proc. of the Eurographics/IEEE-VGTC Symposium on Visualization (EuroVis), 2009.

## Part II: Slides

Contents:	Slide numbers
1. Introduction .....	1–23
2. Information Visualization Theory	
a. Representational theory, cognitive psychology, pre-attentive processing .....	24–63
b. Perception, pre-attentive processing.....	64–70
c. Interaction & animation .....	71–84
d. Assessing and validating visualization .....	85–90
e. Visualization Design and Usability .....	91–97
3. Review of Linguistic Visualization .....	98–110
a. Visualization of non-textual linguistic data .....	111–113
b. Convergence of linguistic and non-linguistic data .....	114–119
c. Document content visualizations .....	120–126
d. Text collections and information retrieval .....	127–134
e. Streaming data .....	135–139
f. Literary analysis .....	140–145
g. Linguistic analysis .....	146–153
h. NLP interfaces .....	154–158
4. Tools for Visualization	
a. Online tools .....	159–166
b. Programming libraries .....	167–176
c. Visualization software .....	177–183
d. Emerging research .....	184–186
5. Open Research Problems .....	187–197

## Part III: Annotated Bibliography

# CHAPTER 1



**Graphics is the visual means of resolving logical problems.**

- Bertin (1977/1981, p. 16)

# Information Visualization

To understand something is called “seeing” it. We try to make our ideas “clear,” to bring them into “focus,” to “arrange” our thoughts. The ubiquity of visual metaphors in describing cognitive processes hints at a nexus of relationships between what we see and what we think. When we imagine someone hard at mental work, we might picture a scholar drawing a diagram, a book of sources open at her side. Or we might imagine a stockbroker, watching computer displays of financial data, rushing to act on events. Whatever the activity, mental work and perceptual interactions of the world are likely to be interwoven.

This interweaving of interior mental action and external perception (and manipulation) is no accident. It is the essence of how we achieve expanded intelligence. As Norman says,

*The power of the unaided mind is highly overrated. Without external aids, memory, thought, and reasoning are all constrained. But human intelligence is highly flexible and adaptive, superb at inventing procedures and objects that overcome its own limits. The real powers come from devising external aids that enhance cognitive abilities. How have we increased memory, thought, and reasoning? By the invention of external aids: It is things that make us smart. (Norman, 1993, p. 43)*

An important class of the external aids that make us smart are graphical inventions of all sorts. These serve two related but quite distinct purposes. One purpose is for communicating an idea, for which it is sometimes said, “A picture is worth ten thousand words.”<sup>1</sup> Communicating an idea requires, of course, already having the idea to communicate. The second purpose is to use graphical means to create or discover the idea itself: using the special properties of visual perception to resolve logical problems, as Bertin (1977/1981) would say. *Using vision to think*. This second sense of graphics is the subject of this book.

Graphic aids for thinking have an ancient and venerable history. What is new is that the evolution of computers is making possible a medium for graphics with dramatically improved rendering, real-time interactivity, and dramatically lower cost. This medium allows graphic depictions that

automatically assemble thousands of data objects into pictures, revealing hidden patterns. It allows diagrams that move, react, or even initiate. These, in turn, create new methods for amplifying cognition, new means for coming to knowledge and insight about the world. A few years ago, the power of this new medium was applied to science, resulting in scientific visualization. Now it is possible to apply the medium more generally to business, to scholarship, and to education. This broader application goes under the name of *information visualization*. The purpose of this book is to introduce information visualization, to collect some of the important papers in the field, and to give samples of some of the latest work.

## EXTERNAL COGNITION

To understand the intuition behind information visualization, it is useful to gain an appreciation for the important role of the external world in thought and reasoning. This notion is sometimes called *external cognition* (Scaife and Rogers, 1996) to express the way in which internal and external representations and processing weave together in thought. As Norman suggests, the use of the external world, and especially the use of cognitive artifacts or physical inventions to enhance cognition, is all around us.

### Multiplication Aids

Take multiplication, one of the most mental of activities. Have a person multiply a pair of two-digit numbers, such as  $34 \times 72$ , in his or her head and time how long it takes. Now repeat the experiment with another pair of numbers, in longhand using pencil and paper.

$$\begin{array}{r} 34 \\ \times 72 \\ \hline 68 \\ 23^2 80 \\ \hline 24^1 48 \end{array}$$

<sup>1</sup>According to Paul Martin Lester, professor of communications at the University of California at Fullerton, this quotation was simply made up by ad writer Frederick R. Barnard and included as an invented “Chinese proverb” in a streetcar advertisement for Royal Baking Powder. The ad writer wanted to make the point that pictures can attract attention faster than other media. See <http://www5.fullerton.edu/les/ad.html> and *Printers’ Ink*, March 10, 1927.



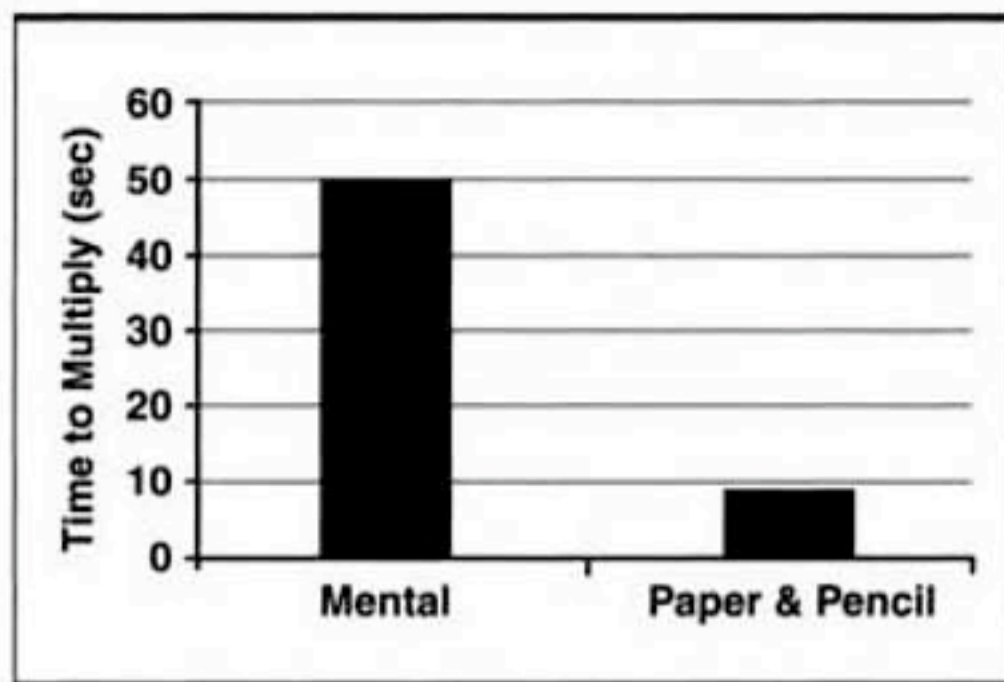


FIGURE 1.1

Use of external aids amplifies ability to do multiplication.

Figure 1.1 shows the result of trying this experiment on a hapless colleague: pencil and paper reduced the time by a factor of five. (Too keep the story simple, we made sure that none of the digits was 0 or 1 and that the colleague did not know the Trachtenberg or other special system for mental multiplication). As this informal demonstration shows, visual and manipulative use of the external world amplifies cognitive performance, even for this supposedly mental task. And if we had chosen to multiply 3- or 4-digit numbers—or 25-digit numbers—then the task would have quickly become impossible to do mentally at all (at least without special methods).

Why does using pencil and paper make such a difference? Quite simply, mental multiplication is not itself difficult. What is difficult is holding the partial results in memory until they can be used. The visual representation, by

holding partial results outside the mind, extends a person's working memory. Applying this principle backwards, people can learn apparently astonishing feats of mental arithmetic by learning special algorithms like the Trachtenberg system that minimize internal working memory (Cutler and McShane, 1960). The cost is in the extra effort to learn the algorithms.

Manipulable, external visual representations like long-hand arithmetic with paper and pencil work a different way from the algorithmic tricks. By writing intermediate results in neatly aligned columns (plus little numbers for carries), the doer of multiplication creates a visual addressing structure that minimizes visual search and speeds access. An internal memory task is converted to an external visual search and manual writing task.

External visual representations for multiplication can work in other ways as well. The slide rule is an analogue interactive visual device that represents quantities as scales with length proportional to their logarithms. Sliding the scales adds these lengths and hence multiplies the quantities (Figure 1.2). Instead of aiding cognition by extending working memory, the slide rule actually does the visual computation (except for placing the decimal point). There are no partial results at all. Slide rules are devices for interactive manipulation of good visual representations.

Nomographs are visual devices that allow specialized computations. The nomograph in Figure 1.3 allows visual calculations and trade-offs for the design of a water conduit. Water needs to be conveyed from a storage pond to a powerhouse by a ditch or a pipe. At the powerhouse, it will be converted to mechanical rotational energy and then to electric energy. The ditch will absorb some of the energy from the water. Suppose we want to know what slope to give a

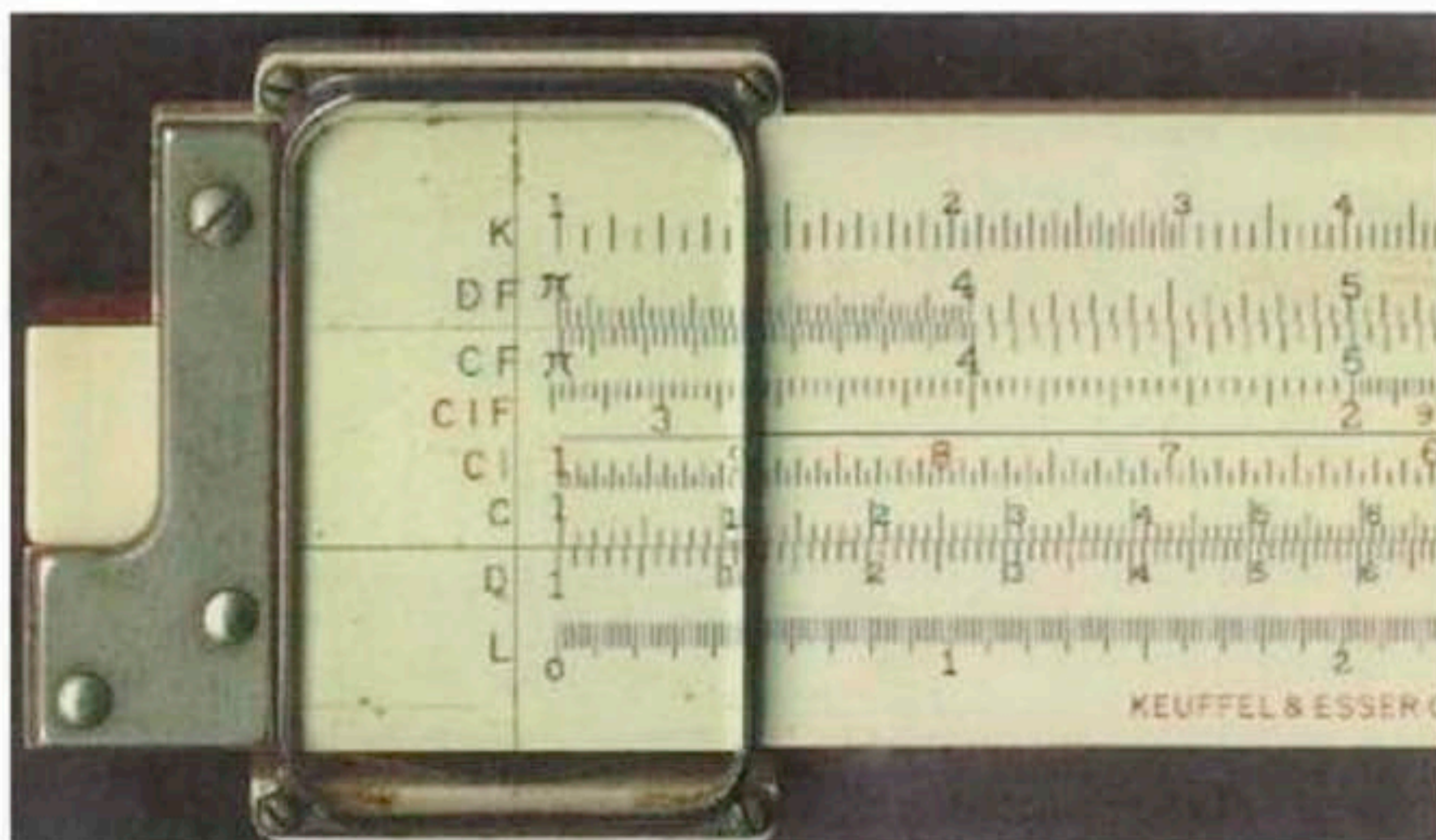


FIGURE 1.2

Section of a slide rule.

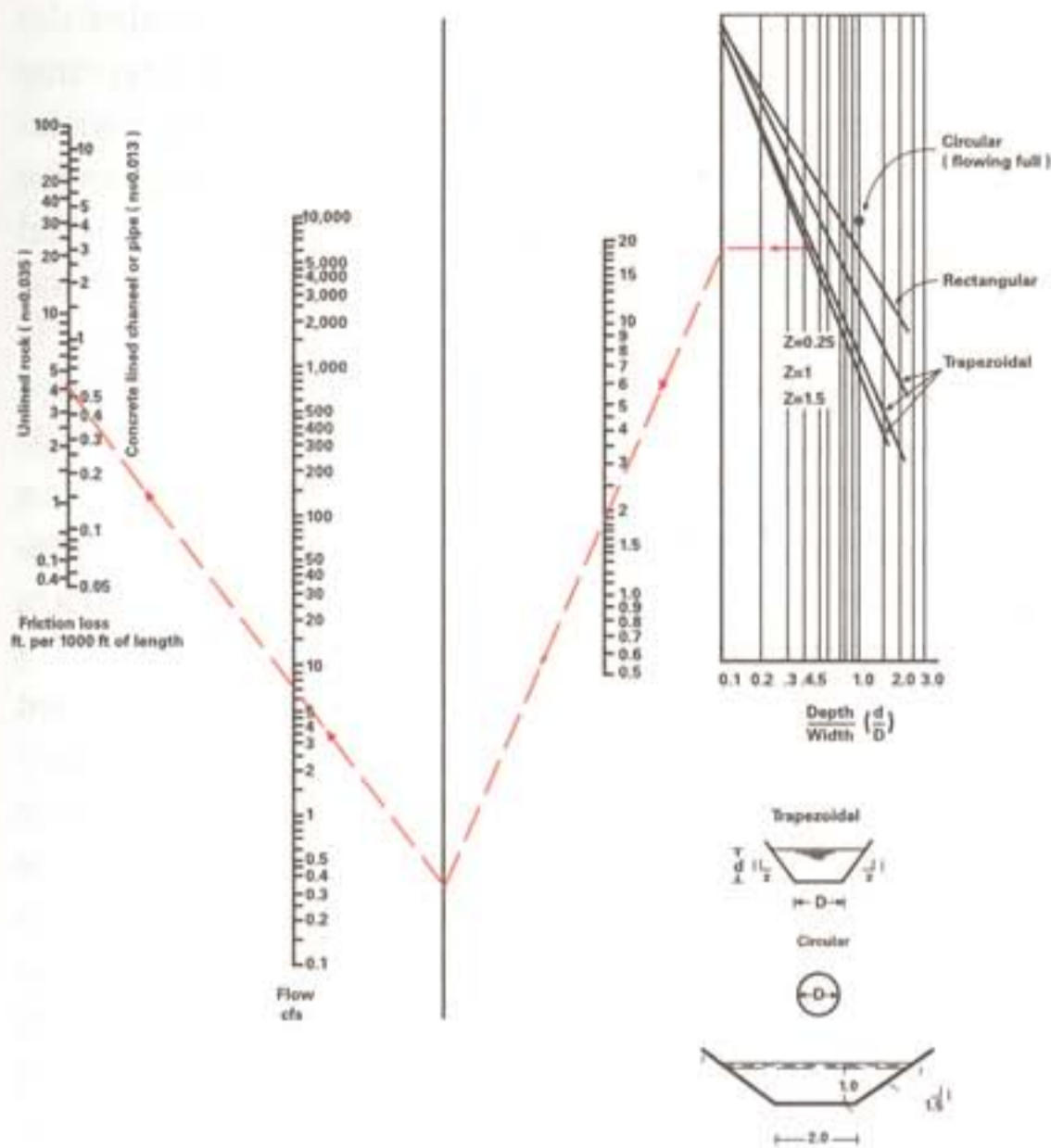


FIGURE 1.3

Nomograph for determining friction loss in a conduit (Leckie et al., 1975, p. 66).

trapezoidal rock ditch in order to overcome frictional losses and deliver 7 cubic ft/sec to the powerhouse. The base of the ditch is  $D = 2$  ft. Its sides are inclined at 1:1.5. Water is to be carried at a depth of  $d = 1$  ft. We use the nomograph as follows:

1. On the right side of the nomograph, we locate the point corresponding to a ratio of  $d/D = 1/2 = 0.5$  and the line  $Z = 1.5$  for the slope of the sides of the ditch.
2. With a ruler, we determine a line between that point and  $D = 2$  ft on the next scale. This determines a point on the Center Reference Line of the diagram.
3. We now use that point and the required flow rate of 7 cfs on the Flow cfs scale to determine a new line.
4. We read our answer on the Friction Loss scale of about 4 ft drop/1000 ft of ditch length, which equals 0.4% slope.

We could easily do “what if” calculations, just by adjusting slightly the position of the ruler. What happens if we make the ditch rectangular? if we use a pipe? if our requirements for flow are changed? This reasoning, trivial with the nomograph, would be difficult to do in the head (unless you were a specialist) or even with a calculator.

Slide rules were superseded as computational devices by pocket calculators. The lesson is that although visually based devices can aid mental abilities, they are not the only means of augmentation. Direct computational devices may do as well or better. But then the direct computational devices may themselves become a component of an even more powerful visually based system. An example is the Graphing

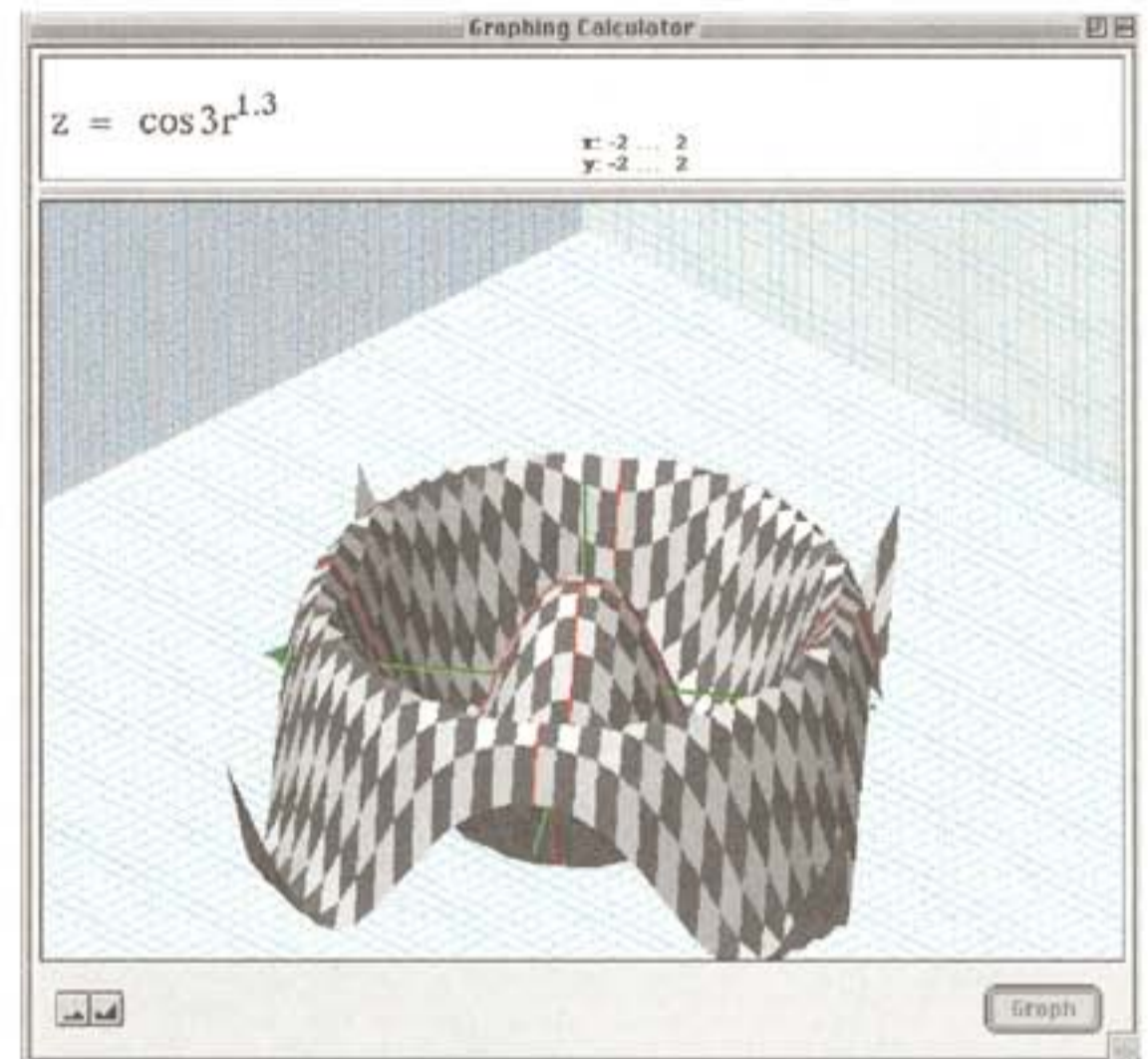


FIGURE 1.4

The Apple Graphing Calculator.

Calculator (Avitzur, Robins, and Newman, 1994). In Figure 1.4, the user has typed in a simple trigonometric formula to evaluate  $z = \cos 3r^{1.3}$ . Instantly, a visualization is displayed involving perhaps millions of computations of the sort that would be done by a slide rule or a simple pocket calculator. The user could not quickly absorb this many calculations. Figure 1.4, on the other hand, produces insight that occasionally surprises even people with some mathematical sophistication. The visualization is designed with skill. The muted background provides orientation. Lighting is used to give the different axes identity. The graph itself uses a checkered pattern and lighting effects that enhance contours. The user can set the figure into spinning animation, highlighting the 3D effect and revealing the figure from different angles. If the number 3 in the formula is replaced by  $n$ , a slider control appears. The slider can vary  $n$ , showing its effect on the graph. The slider can even be put into automatic animation.

### Navigation Charts

Let us consider another example of a visual aid to cognition, navigating at sea. Virtually all computations of a ship's position are done using a nautical chart (see Hutchins, 1996) of the sort shown in Figure 1.5. The chart is a navigator's main representation of position, even though the chart shows a view that no navigator ever sees. In fact, because the earth is round and it is convenient to use flat charts, compromised projections of the round earth on the flat chart must be used such that graphical operations performed on the charts will work.

A navigation chart is really a sort of visual analogue computing device for navigation. With the chart, the navigator can compute a ship's compass heading to its destination if the destination is not too far. If the trip is long, however, a



FIGURE 1.5

Navigation chart in use (Hutchins, 1996, Figure 1.3).

constant heading becomes a spiral around the pole. A Mercator projection transforms this spiral back into a straight line. But radio beacons and the shortest line to distant points follow a great circle route, which is not a straight line on either projection. A straightedge ruler can, however, be used to plot a great circle route as a straight line on a Lambert projection. Each type of map sacrifices accurate representation of some physical property of the earth, because its true purpose is to support specific calculations. Of course, irregular features on the earth's surface can modify a straight route: coastline shapes, ocean depths, political ownership of territory, navigational beacons. The map is not just a calculator but also a storage device, storing for access enormous amounts of information about the earth's irregular features naturally located near where they are needed for calculation.

### Diagrams

Diagrams are another important class of visual aids, although they are usually not interactive. Diagrams can lead to great insight, but also to the lack of it. Tufte (1997) cites as an example the accident of the space shuttle *Challenger*. There was a question whether the shuttle should be launched on a cold day. The decision depended on whether

the temperature would make the O-rings that sealed the sections of the booster rockets unsafe. Figure 1.6 reprints one of the diagrams used for this decision by the booster rocket manufacturer to analyze earlier launch damage to the booster seals. On the chart, boosters are shown in historical order of launch. The choice of presentation obscures the important variables of interest: temperature is shown textually rather than graphically; degree of damage is not mapped onto a natural graphical scale (and there is no legend). Diagrams of the rockets clutter the chart, making other patterns difficult to see. Consequently, the diagram reveals no obvious patterns. It seems to show that the incidents of damage are relatively few.

Tufte's chart of the same data (Figure 1.7) tells a different story. It uses a simple scattergraph depicting the relationship between the two major variables of interest. Different types of damage are combined into a single index of severity. The proposed launch temperature is also put on the chart to show it in relation to the data. The diagram reveals a clear pattern of damage for launches below 65°. In fact, the new diagram shows that there was always damage below 65° and that the most serious damage occurred at the lowest temperature. It shows that the proposed launch is very much

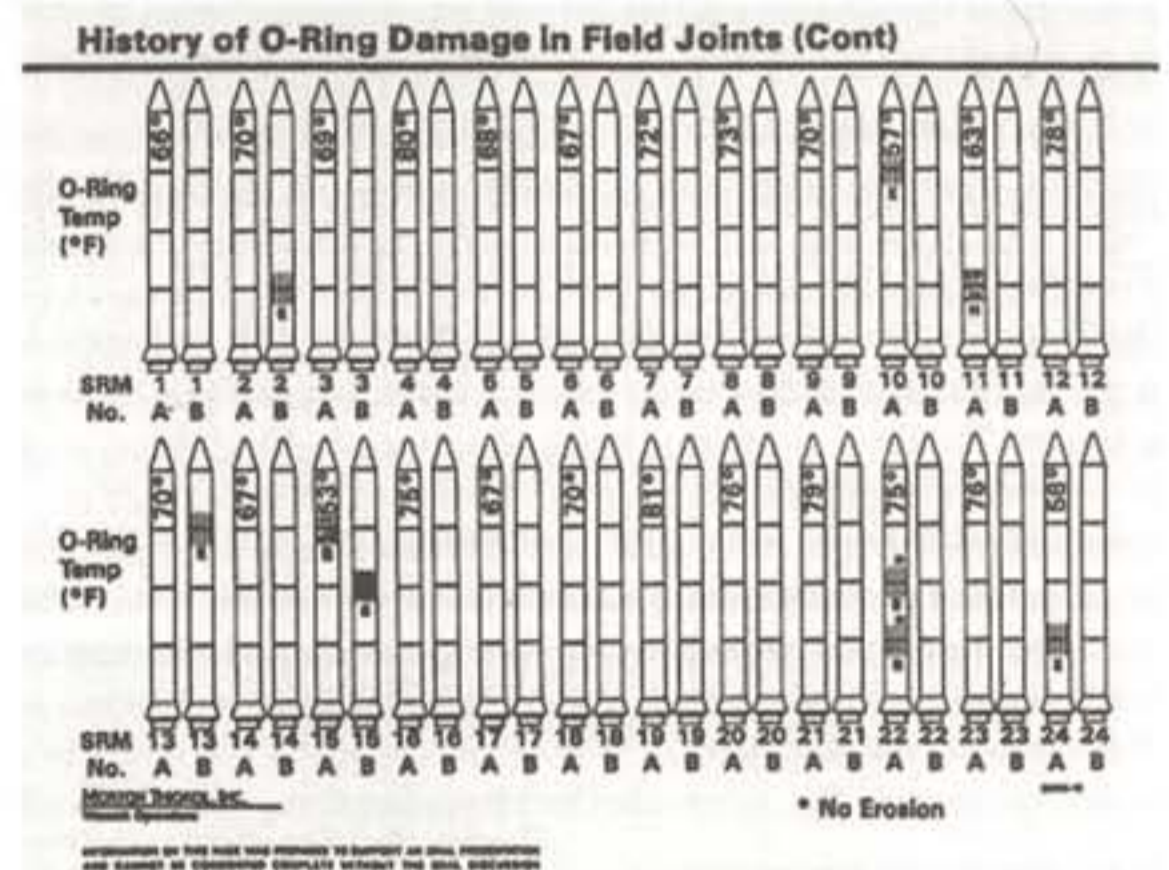


FIGURE 1.6

One of the diagrams of O-ring damage used to make the decision to launch *Challenger* (Nielson, Hagen, and Muller, 1997, vol. v, p. 896).

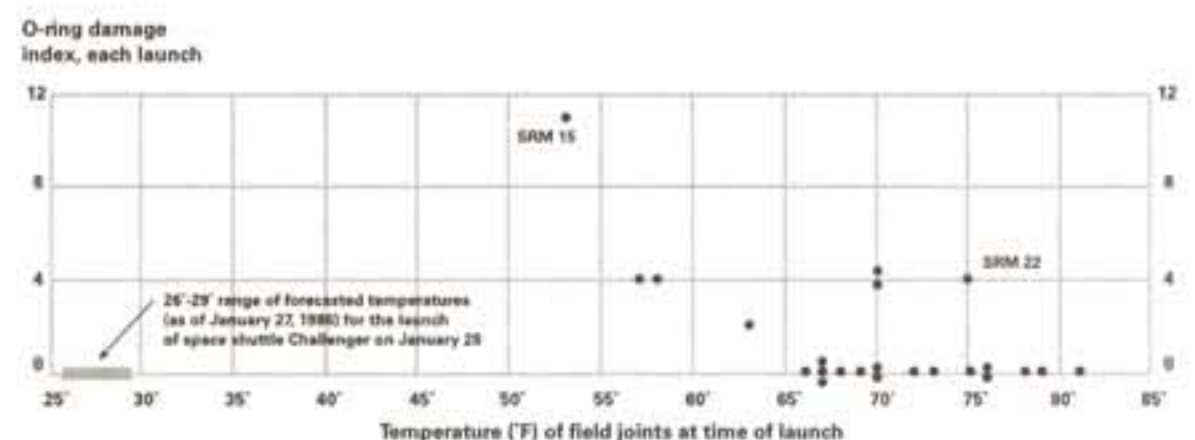


FIGURE 1.7

Scattergraph of O-ring damage index as a function of temperature (Tufte, 1997, p. 45).

colder than this previous lowest temperature. Had the engineers seen this diagram instead of Figure 1.6, it is difficult to believe they would have recommended launch. The diagram illustrates how the right representation of a problem, often the right visual representation, can make a problematic decision obvious. It also illustrates Tufte's point that "There are right ways and wrong ways to show data; there are displays that reveal the truth and displays that do not" (Tufte, 1997, p. 45).

A related but different lesson comes from the next two diagrams. The first of these, Figure 1.8, shows the sleep/wake cycles of a newborn infant (Winfree, 1987). In these diagrams, a good representation reveals surprisingly simple patterns embedded in massive data and great complexity. Each line in Figure 1.8 represents time sleeping, and each dot is a feeding. In the weeks after birth, the sleep cycle shows considerable irregularity, but we can detect the natural 25-hour patterns exhibited by humans when they are isolated from the light/dark cycle of the day. Around the 17th week, the infant's sleep/wake cycle synchronizes with the 24-hour solar day. The diagram presents every one of some three million observations, yet allows the large-scale pattern to be detected.

The second diagram, Figure 1.9, shows another time cycle aggregated from massive data and calculations. Tides at any given point on earth generally have a cycle of around 12 h 26 m. A more complex picture emerges if we ask what are all of the points on the earth that are in the same tide phase at a given time. High tide cannot be everywhere at

once, since there is only a fixed amount of water in the ocean. While some places on earth are in high tide, others must be in low tide or in between the two. The figure plots the tidal phase of each point of earth relative to Greenwich, England, by mapping tidal phase onto the color wheel (used because the color wheel is circularly continuous without a zero point). The figure reveals the surprising existence of singularities called *anphidromic points*, points at which there are no tides at all. *Cotidal lines* (contour lines consisting of points at the same tide phase) circulate around these anphidromic points, some clockwise, some counterclockwise. The diagram makes it possible to comprehend this phenomenon, which is unintuitive and made more complicated by the irregular shape of the earth's landmasses.

As our brief examination illustrates, visual artifacts aid thought; in fact, they are completely entwined with cognitive action. The progress of civilization can be read in the invention of visual artifacts, from writing to mathematics, to maps, to printing, to diagrams, to visual computing. As Norman says, "The real powers come from devising external aids that enhance cognitive abilities." Information visualization is about just that—exploiting the dynamic, interactive, inexpensive medium of graphical computers to devise new external aids enhancing cognitive abilities. It seems obvious that it can be done. It is clear that the visual artifacts we have discussed so far have profound effects on peoples' abilities to assimilate information, to compute with it, to understand it, to create new knowledge. Visual artifacts and computers do for the mind what cars do for the feet or steam

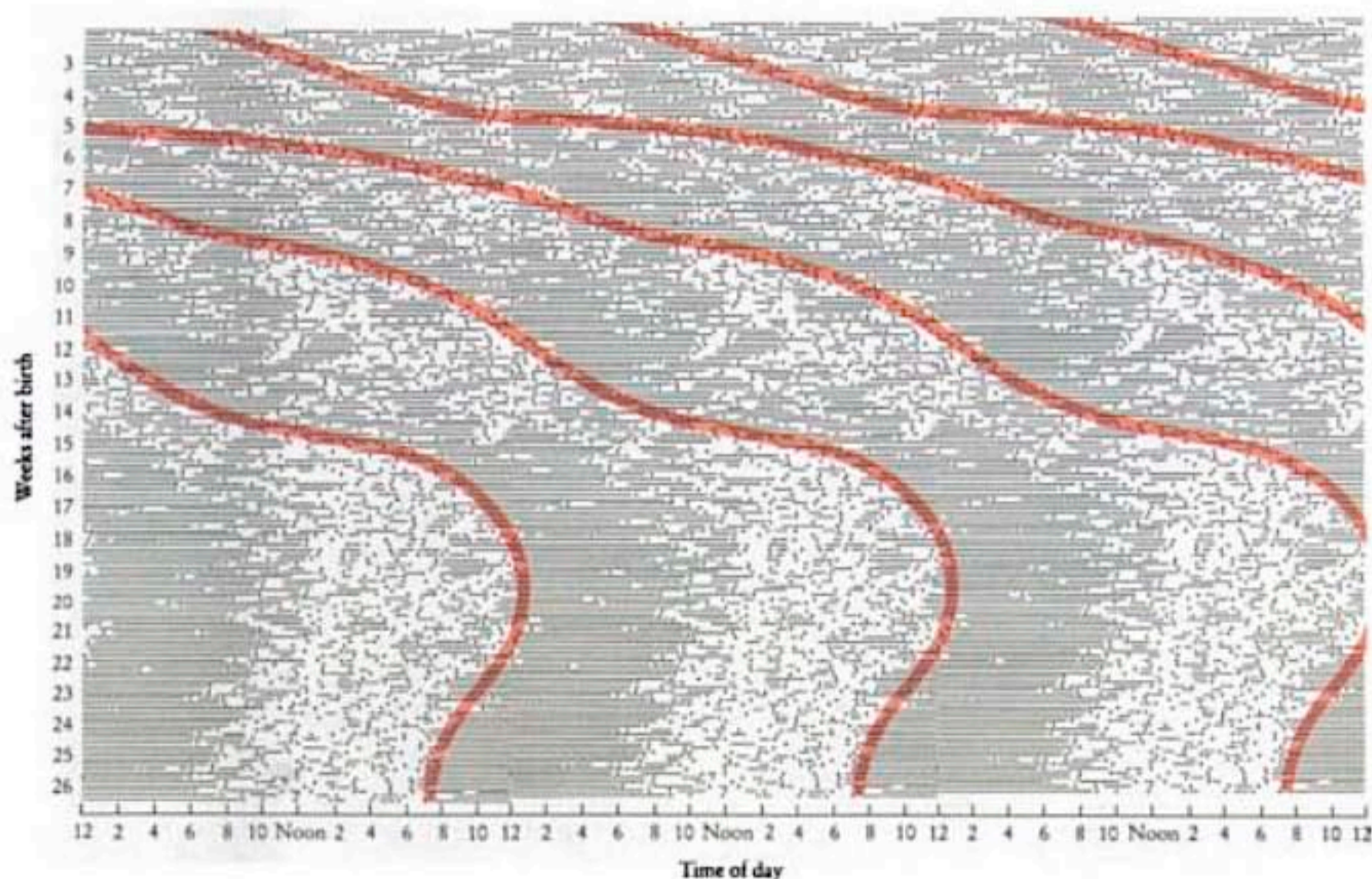


FIGURE 1.8

Sleep/wake cycles of a newborn infant. To make the cycles easier to see, each line starts a new day, but three days are plotted on each line. The infant transitions from the natural human 25-hour cycle at birth to the 24-hour solar day (Winfree, 1987, p. 31).

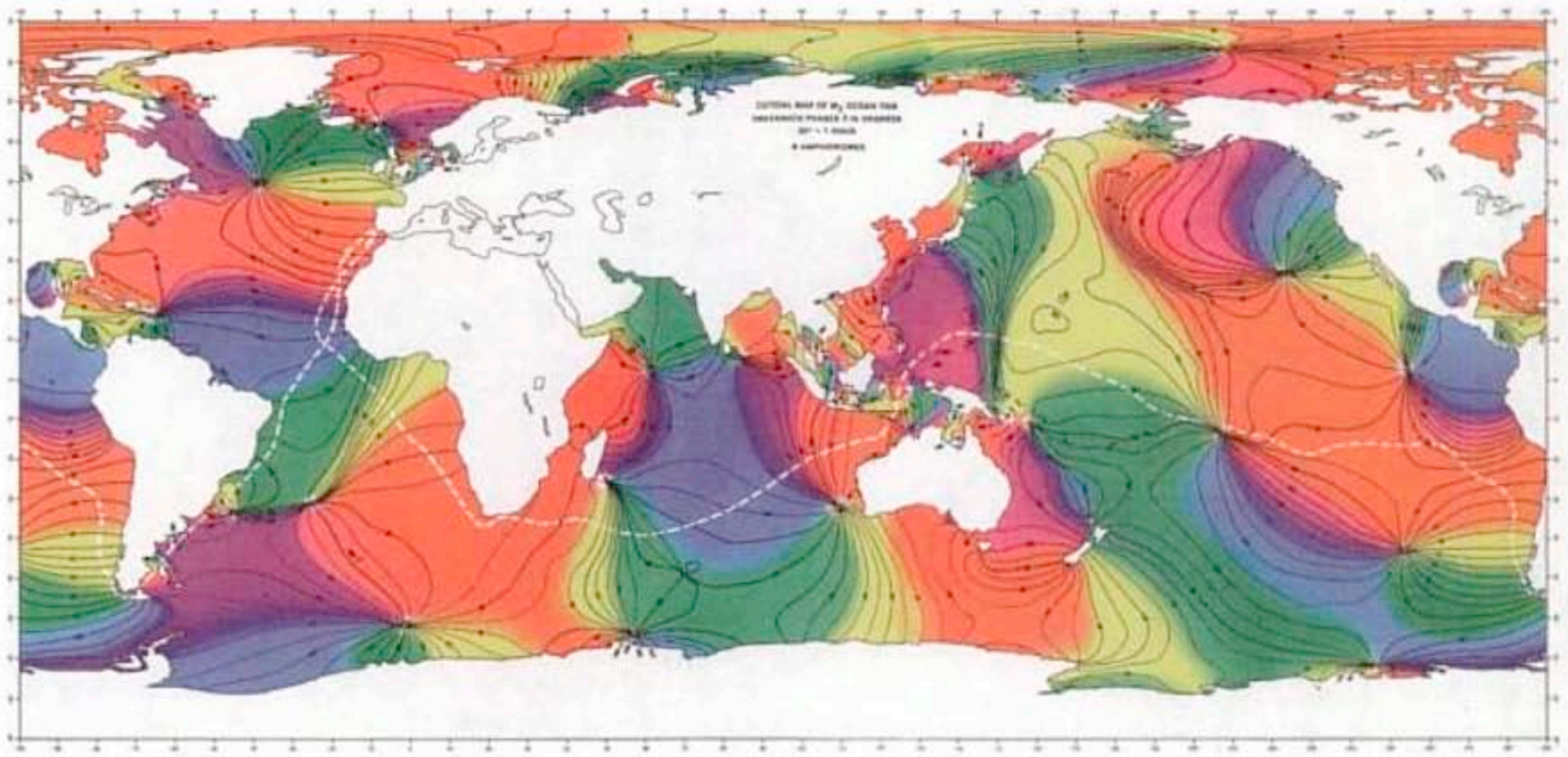


FIGURE 1.9

Cotidal chart. Tide phases relative to Greenwich are plotted for all the world's oceans. Phase progresses from red to orange to yellow to green to blue to purple. The lines converge on anfidromic points. The dotted white line shows the route of Magellan's ship (Winfrey, 1987, p. 17).

shovels do for the hands. But it remains to puzzle out through cycles of system building and analysis how to build the next generation of such artifacts.

## INFORMATION VISUALIZATION

Several activities are concerned with the creation of visual artifacts, and we need to disentangle their relationships in order to set information visualization in context. Let us start with the notion of visualization itself, which we define as follows:

### VISUALIZATION :

The use of computer-supported, interactive, visual representations of data to amplify cognition.

Cognition is the acquisition or use of knowledge. This definition has the virtue of focusing as much on the purpose of visualization as the means. Hamming (1973) said, "The purpose of computation is insight, not numbers." Likewise for visualization, "The purpose of visualization is insight, not pictures." The main goals of this insight are *discovery*, *decision making*, and *explanation*. Information visualization is useful to the extent that it increases our ability to perform these and other cognitive activities.

Visualization dates as an organized subfield from the NSF report, *Visualization in Scientific Computing* (McCormick and DeFanti, 1987). There it is conceived as a tool to permit handling large sets of scientific data and to enhance scientists' ability to see phenomena in the data. Although it is not a necessity of the original conception, scientific visualiza-

tions tend to be based on physical data—the human body, the earth, molecules, or other. The computer is used to render visible some properties. While visualizations may derive from abstractions on this physical space, the information is nevertheless inherently geometrical. For example, in Figure 1.10, a visualization of ozone concentration in the atmosphere, the visualization is based on a physical 3D representation of the earth. In Figure 1.11, a visualization of fluid flow around a hemispherical surface, the colors of the tubes show changes in the eigenvector of the stress tensor of flow.

Both of these visualizations show abstractions, but the abstractions are based on physical space. Nonphysical infor-

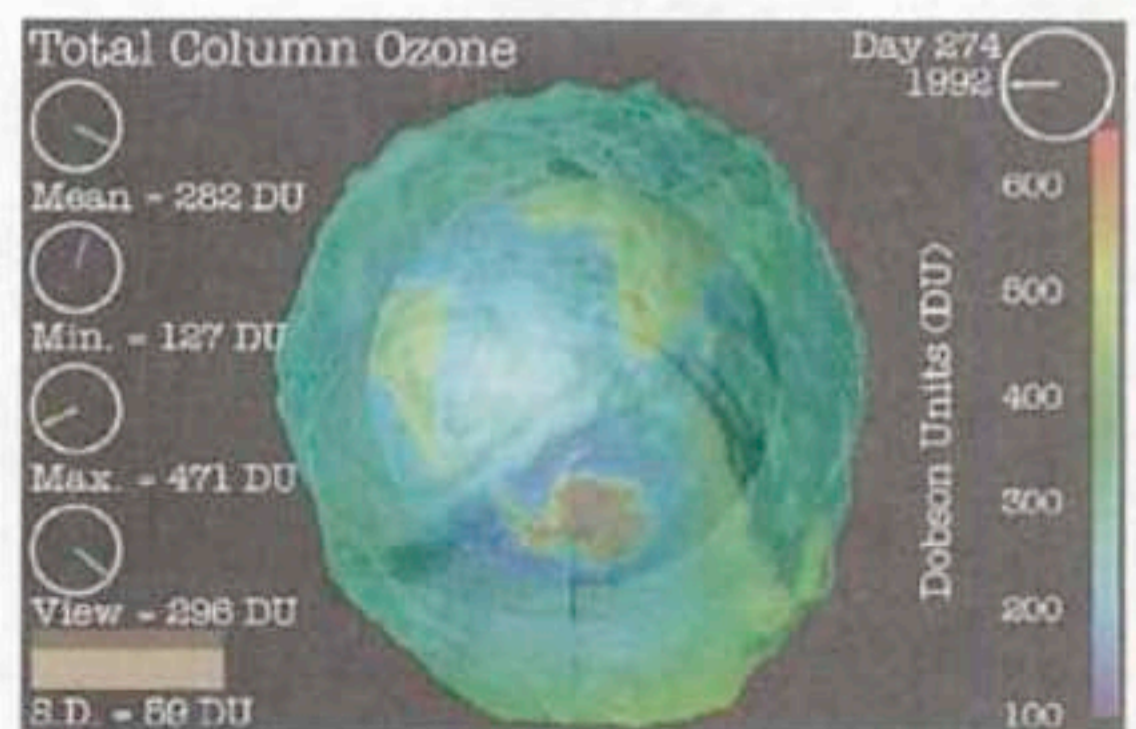


FIGURE 1.10

Ozone layer surrounding earth. L. Treinish, IBM. Used with permission.

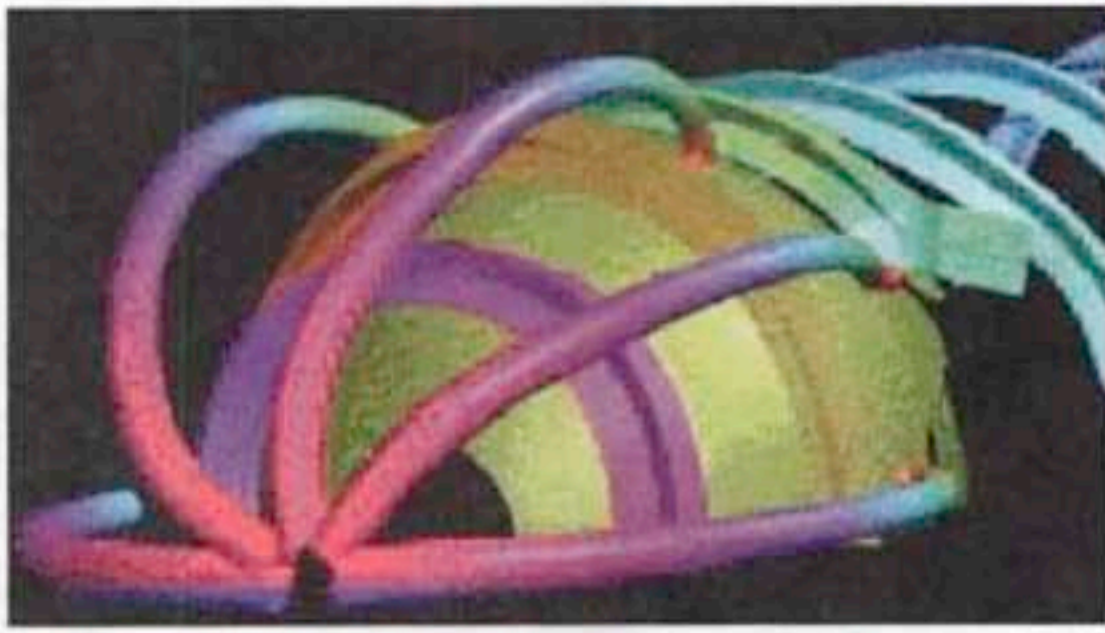


FIGURE 1.11

Stress tensor in a flow past a hemisphere cylinder (Lavin, Levy, and Hesselink, 1997, Figure 5).

mation—such as financial data, business information, collections of documents, and abstract conceptions—may also benefit from being cast in a visual form, but this is information that does not have any obvious spatial mapping. In addition to the problem of how to render visible properties of the objects of interest, there is the more fundamental problem of mapping nonspatial abstractions into effective visual form. There is a great deal of such abstract information in the contemporary world, and its mass and complexity are a problem, motivating attempts to extend visualization into the realm of the abstract (Card, Robertson, and Mackinlay, 1991). As we saw before, visual aids to cognition benefit from good visual representations of a problem and from interactive manipulation of those representations. We define *information visualization* as follows:

**INFORMATION VISUALIZATION:**

The use of computer-supported, interactive, visual representations of abstract data to amplify cognition.

In Table 1.1, we have recorded a number of working definitions to clarify the relationships among concepts related to information visualization. *External cognition* is concerned with the interaction of cognitive representations and processes across the external/internal boundary in order to support thinking. *Information design* is the explicit attempt to design external representations to amplify cognition. *Data graphics* is the design of visual but abstract representations of data for this purpose. *Visualization* uses the computer for data graph-

ics. *Scientific visualization* is visualization applied to scientific data, and *information visualization* is visualization applied to abstract data. The reasons why these two diverge are that scientific data are often physically based, whereas business information and other abstract data are often not. It should be noted that while we are emphasizing visualization, the general case is for *perceptualization*. It is just as possible to design systems for *information sonification* or *tactilization* of data as for multiple perceptualizations. Indeed, there are advantages in doing so. But vision, the sense with by far the largest bandwidth, is the obvious place to start, and it would take us too far afield to cover all the senses here.

### Origins of Information Visualization

These distinctions carry with them some of the historical evolution of this area. Information visualization derives from several communities. Work in data graphics dates from about the time of Playfair (1786), who seems to have been among the earliest to use abstract visual properties such as line and area to represent data visually (Tufte, 1983). Starting with Playfair, the classical methods of plotting data were developed. In 1967, Bertin, a French cartographer, published his theory of graphics in *The Semiology of Graphics* (Bertin, 1967/1983; Bertin, 1977/1981). This theory identified the basic elements of diagrams and described a framework for their design. Tufte (1983) published a theory of data graphics that emphasized maximizing the density of useful information. Both Bertin's and Tufte's theories became well known and influential in the various communities that led to the development of information visualization as a discipline.

Although the data graphics community was always concerned with statistical graphics, Tukey (1977) began a movement from within statistics with his work on *Exploratory Data Analysis*. The emphasis in this work was not on the quality of the graphics but on the use of pictures to give rapid statistical insight into data. For example, "box and whisker" plots allowed an analyst to see in an instant the most important four numbers that characterize a distribution. Rocking displays allowed an analyst to see 3D scatterplots without special glasses. Cleveland and McGill (1988) wrote an influential book, *Dynamic Graphics for Statistics*, explicating new visualizations of data in this area. A problem of particular interest was how to visualize data sets with many variables. Inselberg's parallel coordinates method (Inselberg

TABLE 1.1

#### Definitions.

External Cognition	Use of the <i>external world</i> to accomplish cognition.
Information design	Design of <i>external representations</i> to amplify cognition.
Data graphics	Use of <i>abstract, nonrepresentational</i> visual representations of data to amplify cognition.
Visualization	Use of <i>computer-based, interactive</i> visual representations of data to amplify cognition.
Scientific visualization	Use of interactive visual representations of <i>scientific data</i> , typically <i>physically based</i> , to amplify cognition.
Information visualization	Use of interactive visual representations of <i>abstract, nonphysically based data</i> to amplify cognition.

and Dimsdale, 1990) and Mihalisin's technique of cycling through variables at different rates (Mihalisin, Timlin, and Schwegler, 1991 •) were important contributions here. Eick's group worked on statistical graphics techniques for large-scale sets of data associated with important problems in telecommunications networks and in large computer programs (Becker et al., 1995 •; Eick, Steffen, and Sumner, 1992 •). The emphasis of the statisticians was on the analysis of multidimensional, multivariable data and on novel sorts of data.

In 1985, NSF launched an important new initiative on scientific visualization (McCormick and DeFanti, 1987). The first IEEE Visualization Conference was in 1990. This community was led by earth resource scientists, physicists, and computer scientists in supercomputing. Satellites were sending back large quantities of data, so visualization was useful as a method to accelerate its analysis and to enhance the identification of interesting phenomena. It was also promising as part of an effort to replace expensive experiments by computational simulation (e.g., for wind tunnels).

Meanwhile, there was interest by the computer graphics and artificial intelligence communities in automatic presentation, the automatic design of visual presentations of data. The effort was catalyzed by Mackinlay's thesis APT (Mackinlay, 1986a), which formalized Bertin's design theory, added psychophysical data, and used it to generate presentations. Roth and Mattis (1990) built a system to do more complex visualizations, such as some of those from Tufte. Casner (1991) added a representation of tasks. The concern for this community was not so much in the quality of the graphics as in automating the match between data types, communication intent, and graphical representations of the data.

Finally, the user interface community saw advances in graphics hardware opening the possibility of a new generation of user interfaces. These interfaces focused on user interaction with large amounts of information, such as multivariate databases or document collections. The first use of the term "information visualization" to our knowledge was in Robertson, Card, and Mackinlay (1989). Feiner and Beshers (1990b) presented a method, worlds within worlds, for showing six-dimensional financial data in immersive virtual reality. Shneiderman (1992b) developed a technique called *dynamic queries* for interactively selecting subsets of data items and treemaps, a space-filing representation for trees. Card, Robertson, and Mackinlay presented ways of using animation and distortion to interact with large data sets in a system called the *Information Visualizer* (Card, Robertson, and Mackinlay, 1991; Robertson, Mackinlay, and Card, 1991; Mackinlay, Robertson, and Card, 1991). The concern was again not so much the quality of the graphics as the means for cognitive amplification. Interactivity and animation were more important features of these systems.

These initial forays were followed by refinements and new visualizations, the different communities mutually influencing each other.

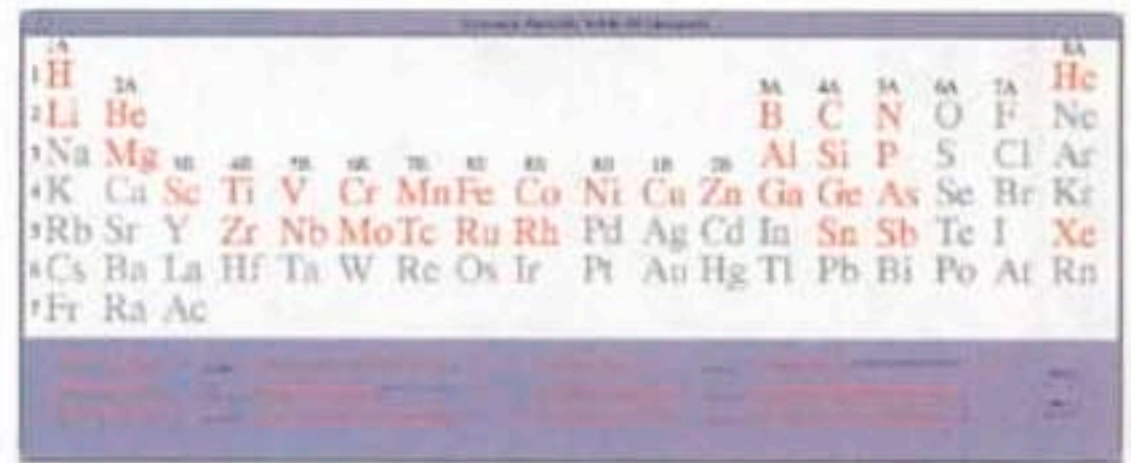


FIGURE 1.12

Periodic table with dynamic queries sliders (Ahlberg, Williamson, and Shneiderman, 1992, Figure 2).

### Active Diagrams

Let us consider some examples of information visualization to make clear what we mean. Our first example amplifies the effect of a good visual representation by making it interactive. The periodic table, created by Mendeleev, is an important diagram in the development of chemistry. In the periodic table, elements are arranged by the number of protons in the atomic nucleus. The way the table is broken into rows and its nonrectangular appearance result from the order in which electrons populate electron subshells. Many physical and chemical properties, such as boiling point and chemical valence, form visual patterns when arranged by the periodic table. In fact, in Mendeleev's lifetime, three elements whose properties were predicted from the periodic table were discovered: gallium, scandium, and germanium (Moore, 1962).

Figure 1.12 shows an information visualization based on the periodic table (Ahlberg, Williamson, and Shneiderman, 1992). The user can set sliders that control which of the elements in the table will be highlighted. For example, the user can indicate interest in ionic radii between 93 and 206 and instantly those values will be highlighted on the table. The sliders can be used to find specific values or to see the trends with the change of some variable. Since the periodic table is already an excellent visual organizer of chemical properties, adding dynamically created patterns on the table is effective.

### Large-Scale Data Monitoring

The second example uses information visualization to monitor and make sense of large amounts of dynamic, real-time data. Figure 1.13 (see Wright, 1995 •) is a depiction of visualization used in a decision-support application. This is an interest-rate, risk-hedging application for a broker-dealer's inventory of fixed-income instruments. The visualization is connected to a real-time database and analytical engine. It replaced 100 screens of rows and columns of numbers in a traditional database reporting system. The visualization shows a thousand bonds arranged by subportfolio along the left and time to maturity along the front. Bonds are shown as vertical

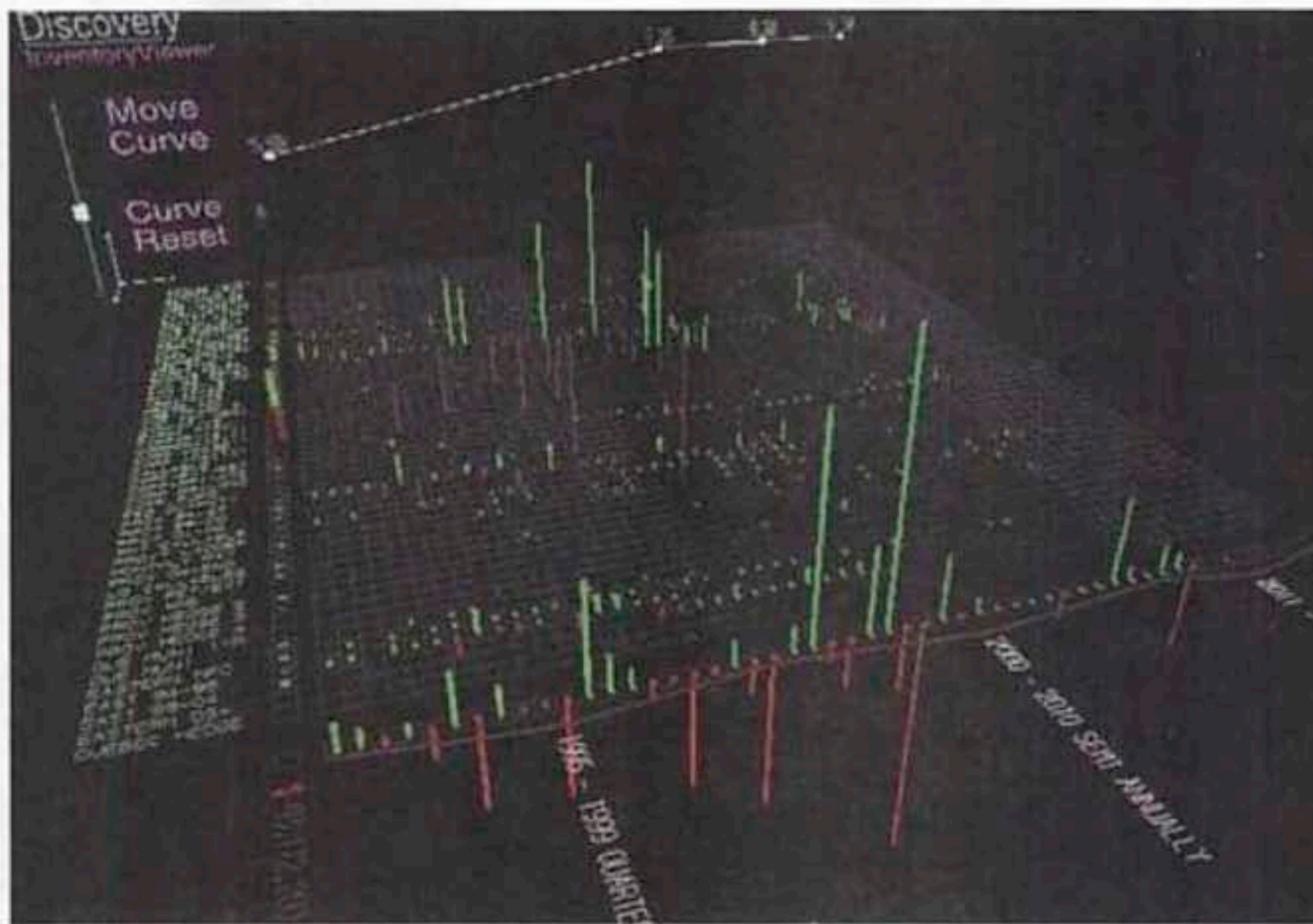


FIGURE 1.13

Positions on the Toronto Stock Exchange. Used by permission of Visible Decisions, Inc.

bars—the higher the bar, the larger the amount of that bond in the portfolio. A total line along the front sums across all subportfolios. Different types of bonds are color coded. At the back is a yield curve. By simply grabbing the yield curve with the mouse and moving it, the user can interactively apply what-if interest rate risk scenarios across the bonds.

Presented as a set of numbers, it would be difficult for a human to monitor these positions and react quickly. Presented visually, it is easy both to spot the items of interest and to tell how these relate to similar stocks or the entire market at a certain point in time. Information visualization is particularly useful for monitoring large amounts of data in real time and under time pressure to make decisions.

### Information Chromatography

Our third example uses a very abstract visualization of real-time data to detect complex new patterns in very large amounts of data: Visualization is used to detect telephone fraud. Figure 1.14 shows a visualization of 40,000 telephone calls, selected by region out of a data set of 20 million international telephone calls. The callers are laid out on a hexagonal grid. Display parameters have been adjusted to call links in a certain frequency range from the call and caller log time histograms in the lower left part of the figure. Figure 1.14 shows the visualization of a set of related calls. By interacting with the set of visualizations, the analyst in this case identified a pattern in which third parties would

route calls from callers in two countries through the United States, charging a fee but then abandoning their phones before paying the bill. Telephone fraud perpetrators change

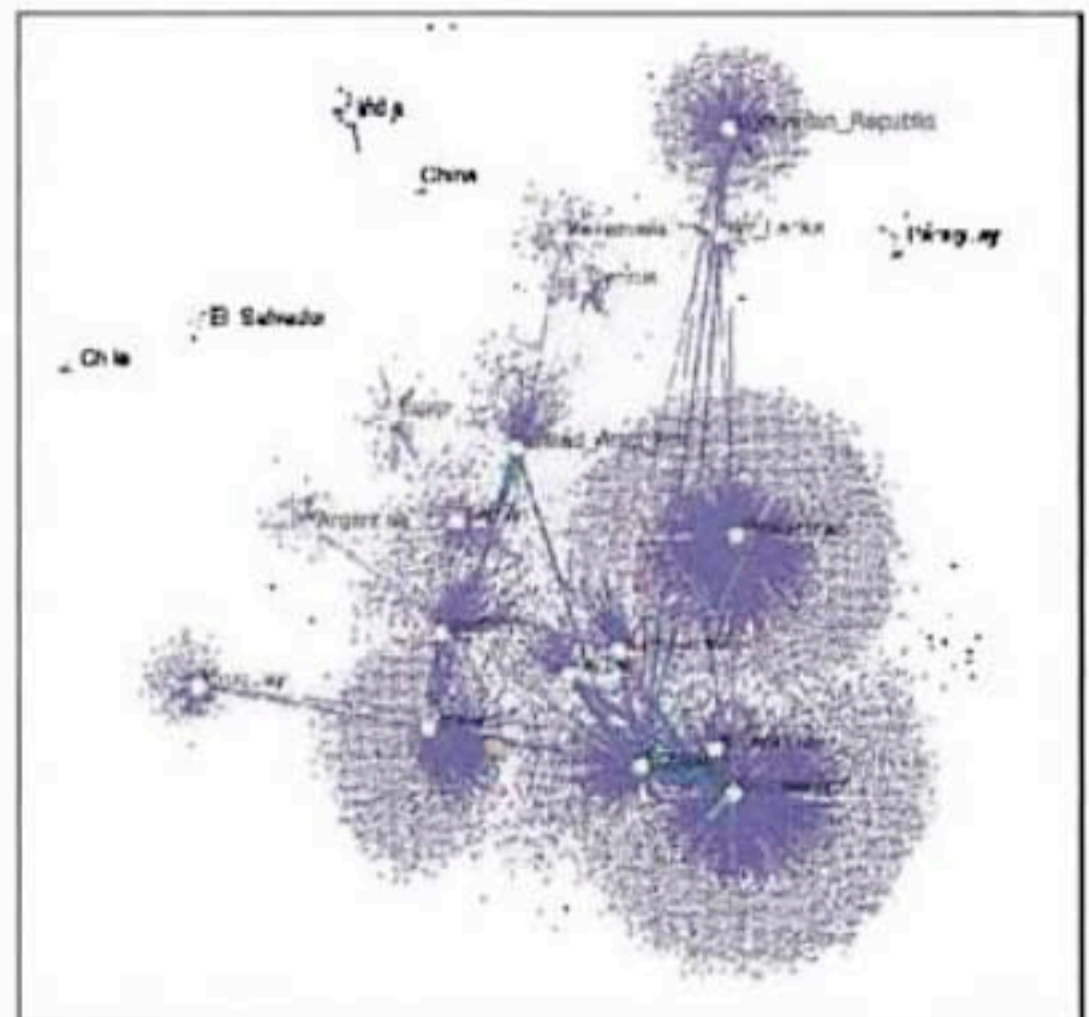


FIGURE 1.14

Visualization used in detecting telephone fraud using Lucent Technologies NicheWorks program (Cox, Eick, and Wills, 1997, Figure 1). Used by permission of Lucent Technologies, Inc.



their patterns of activity frequently to avoid automatic detection algorithms. However, humans with visualization displays are good at picking out new patterns as they occur and thus can respond to changes in the patterns quickly. Information visualization allows human adaptivity to be brought to bear for large data sets under time pressure. We might think of this use as a kind of *information chromatography*: patterns in the data are revealed by laying them out on a particular visual substrate.

The examples of information visualization shown here make use of the power of diagrams, but they add the ability of computers to be interactive and to map large amounts of data into visual forms automatically. As we can see in the examples, the improvement in cognitive performance that occurs can happen for several reasons.

## COGNITIVE AMPLIFICATION

### Knowledge Crystallization

We have said that the purpose of information visualization is to use perception to amplify cognition. Let us give an example of a scenario in which this might happen:

Sue is assigned to buy a laptop computer for a workgroup. If she wishes to make an intelligent choice, it is necessary to understand the purchaser's needs as well as what is on offer in the market. Sue consults the Internet and by a combination of search and browsing acquires documents and data

sets relevant to the purchase. In addition, the purchaser acquires information from colleagues and trade magazines.

The next step is to identify from materials found attributes of interest like processor speed, weight, thickness, and cost—a simple *schema*.

The attributes are laid out in a table: products in rows, features in columns. The table rows and columns are re-ordered and some data is used to make charts. In the process of doing this exercise, the purchaser notices that some machines have interesting new features like high-speed infrared communication and “fire-wire” high-speed communication support for which there is no column. The table is amended with a new column for each of these. The exercise also reveals a lack of information on some of the models. This leads the user to retrieve more information to fill in the table. Using visualizations of table data, the user realizes that the various models represent trade-offs among processing power, multimedia, and portability.

The purchaser then prepares a graphical presentation of two slides to the workgroup presenting the main trade-off (a decision for the group) and the best purchase for each of these trade-offs.

This scenario is an example of a *knowledge crystallization task* (see Figure 1.15). A knowledge crystallization task is one in which a person gathers information for some purpose, makes sense of it (Russell et al., 1993) by constructing a representational framework (which we will refer to as a *schema*), and then packages it into some form for communication or action. The results could be a briefing, a short

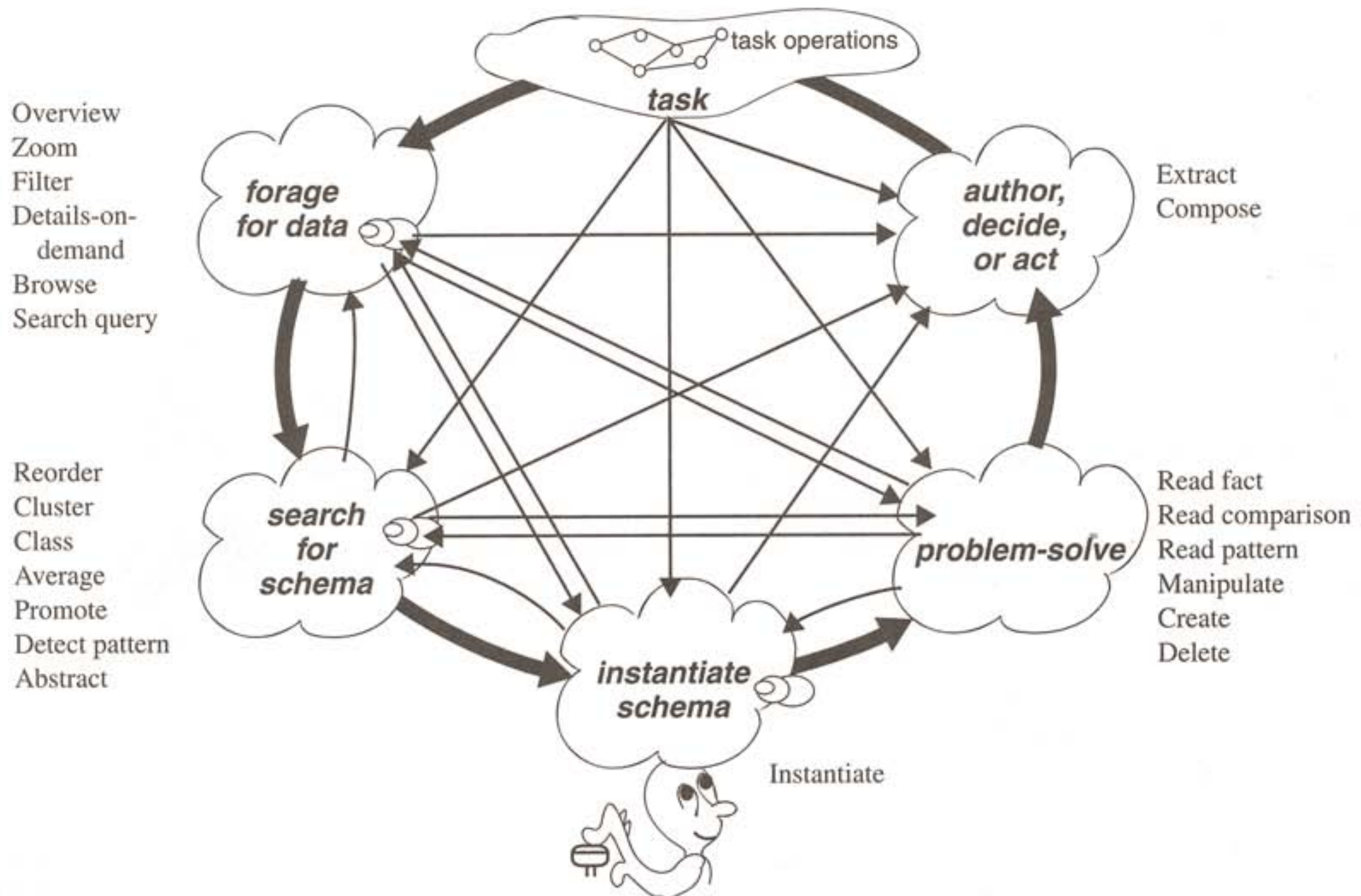


FIGURE 1.15

Knowledge crystallization.

paper, or even a decision or action. Knowledge crystallization tasks are characterized by the use of large amounts of heterogeneous information, ill-structured problem solving, but a relatively well-defined goal requiring insight into information relative to some purpose. Knowledge crystallization tasks are one form of information-intensive work and can themselves be part of more complex forms of knowledge work, such as design. They are an important class of tasks that motivate attempts to develop information visualization.

The preceding scenario has many elements typical of knowledge crystallization as summarized in Figure 1.15. Let's take a closer look at these elements.

- |  |  |
|--|--|
| 1. Information foraging.   | Collecting articles and data on laptop computers.  |
| 2. Search for schema (representation).   | Identification of attributes on which to compare laptops.  |
| 3. Instantiate schema with data. Residue is significant data that do not fit the schema. To reduce residue, go to Step 2 and improve schema. | Make table of laptops $\times$ attributes. Use a "remarks" column to record interesting properties that don't fit into table.  |
| 4. Problem-solve to trade off features.  | Reorder rows and columns of laptop table. Create plots. Delete or mark laptops that are out of the running.  |
| 5. Search for a new schema that reduces the problem to a simple trade-off.   | Cluster into three groups by rearranging the rows in the table, one each for power, multimedia capability, and portability. Within each cluster, delete all but the top one or two machines. |
| 6. Package the patterns found in some output product.  | Create concise briefing on decision for workgroup.   |

Knowledge crystallization involves getting insight about data relative to some task. This usually requires finding some representation (schema) for the data that is efficient for the task. Data are coded in the representation. This encoding leaves residue data that are unencoded or encoded inefficiently. If the residue is too important to ignore, then we search for a better schema. Otherwise, the residual data are omitted. This process of abstraction (that is, schematization) and omission of information is a fundamental principle of how an information processing organism or machine reduces the otherwise unmanageable glut of information to "an amount that can be processed by mental computing equipment with sufficient rapidity to be useful for respond-

ing to changing environmental circumstances" (Resnikoff, 1987, p. 9). As Resnikoff puts it:

*[T]here appears to be a general Principle of Selective Omission of Information at work in all biological information processing systems. The sensory organs simplify and organize their inputs, supplying the higher processing centers with aggregated forms of information which, to a considerable extent, predetermine the patterned structures that the higher centers can detect. The higher centers in their turn reduce the quantity of information which will be processed at later stages by further organization of the partly processed information into more abstract and universal forms. (Resnikoff, 1987, p. 19)*

Information visualization simply abets this process of producing patterns that can be detected and abstracted.

In order to do knowledge crystallization, there must be data, a task, and a schema. If the data are not to hand, then information visualization can aid in the search for it. If there is a satisfactory schema, then knowledge crystallization reduces to information retrieval. If there is not an adequate schema, then information visualization is one of the methods by which one can be obtained.

The HomeFinder (Williamson and Shneiderman, 1992), as shown in Figure 1.16, for instance, allows us to describe home prices directly as a scattergraph on location and by looking at certain ranges of house parameters such as number of bedrooms or price. The mappings of variables into visual forms constitute an initial schema. But out of the interactive examination of the relationships, more expensive and larger houses, say, appear in the NW quadrant of Washington. It is possible to create a more sophisticated description of the housing data than is directly visible at any instant: the relative distribution of luxury apartments and low-cost apartments in the city, where the affluent neighborhoods are, what type of housing suitable for a single person can be found within a 15-minute commute of the Capitol building. This new compact description of the data is a new schema. In principle, we could *reexpress* the data in terms of derived concepts like "type of neighborhood," "housing category," or other concepts discovered in the initial analysis.

Roughly, we want to get the most compact description possible for a set of data relative to some task (Gell-Mann, 1994). The saying "a picture is worth ten thousand words" is a statement claiming a particular compaction ratio (although it does not state the comparison units for the picture or the task). More precisely, what we want is a representation that allows large increases in processing efficiency relative to some task (there may be a trade-off between supporting a single task versus a set of tasks).

Figure 1.15 also shows the subtasks of knowledge crystallization supported by information visualization. This is intended as an approximate and suggestive list, since much research remains to be done to understand the task itself and the effects of information visualization design and user behavior. We have associated subtasks with particular main tasks of knowledge crystallization; however, many of the subtasks could be associated with more than one task.

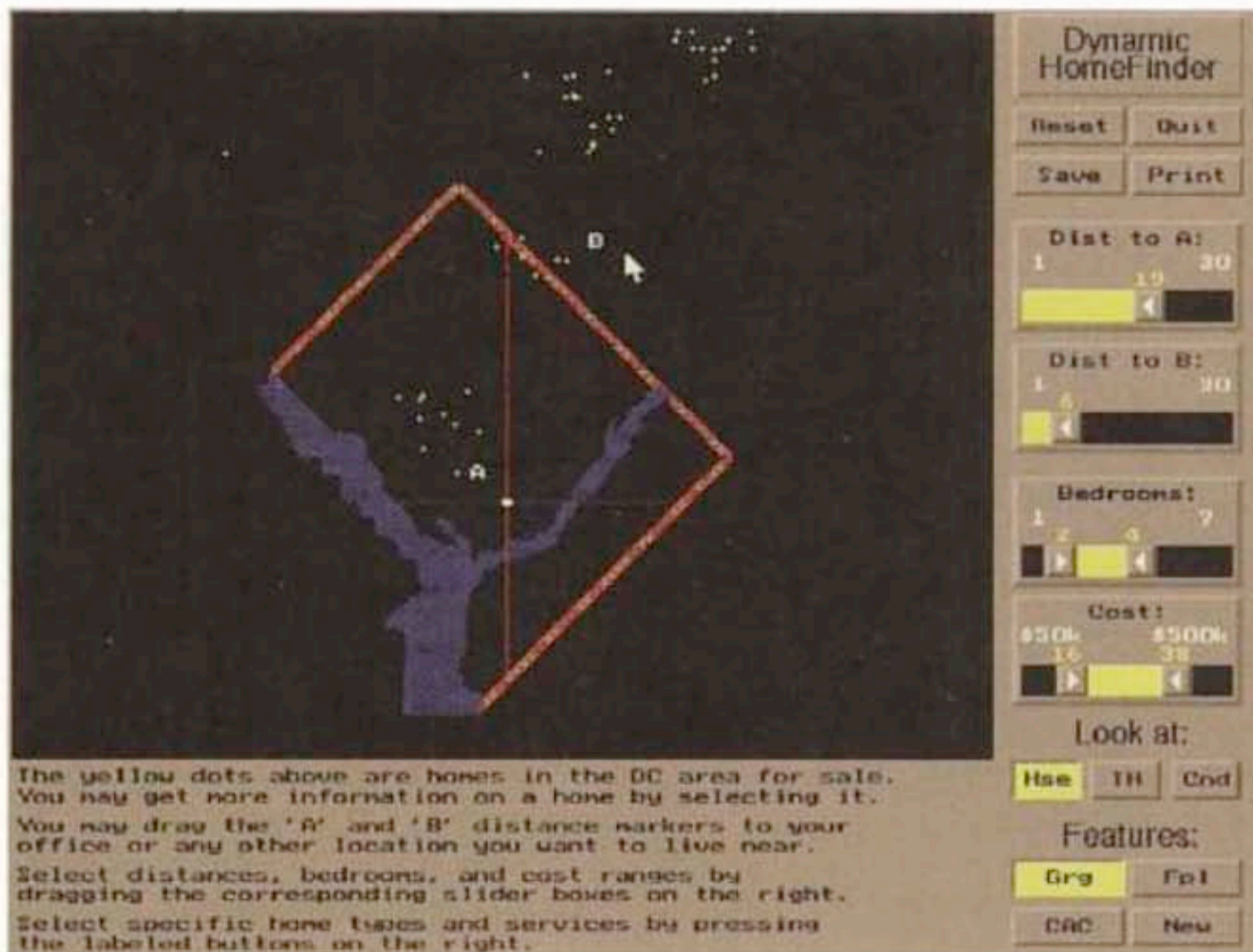


FIGURE 1.16

HomeFinder (Williamson and Shneiderman, 1992). Courtesy of the University of Maryland.

Applying information visualization to knowledge crystallization really means using it to do these different subtasks. Bertin (1977/1981), for example, has called attention to the three levels of “reading” that a diagram can serve. These appear on our diagram as *Read Fact*, *Read Comparison*, and *Read Pattern*. *Read Fact* is visual access to a particular data value—the price of a home, for example. *Read Pattern* uses the whole diagram and picks out the largest-scale pattern—that expensive houses occur in NW Washington, for example. *Read Comparison* is at an intermediate level between these two.

Information visualization can be applied to most parts of knowledge crystallization. To illustrate, a few representative systems are given in Figure 1.17. Figure 1.17(a) shows an attempt to aid foraging by visualizing a portion of the Internet. The diameter of the base represents the number of pages in the site. The height represents the number of other sites pointing to it. The size of the globe represents the number of links to other sites. Figure 1.17(b) shows another aid for foraging by providing a workspace where pages collected from the Web can be arranged and grouped. To help search for a schema, Figure 1.17(c) shows clustering of retrieved data. Figure 1.17(d) shows a table visualization tool that can be used to instantiate a schema and to manipulate cases and variables as part of problem solving. Figure 1.17(e) shows a database visualization tool being used to find logistics resources for emergency planning. Figure 1.17(f) shows a human body made up of many thin slices,

each individually photographed and indexed and available for retrieval.

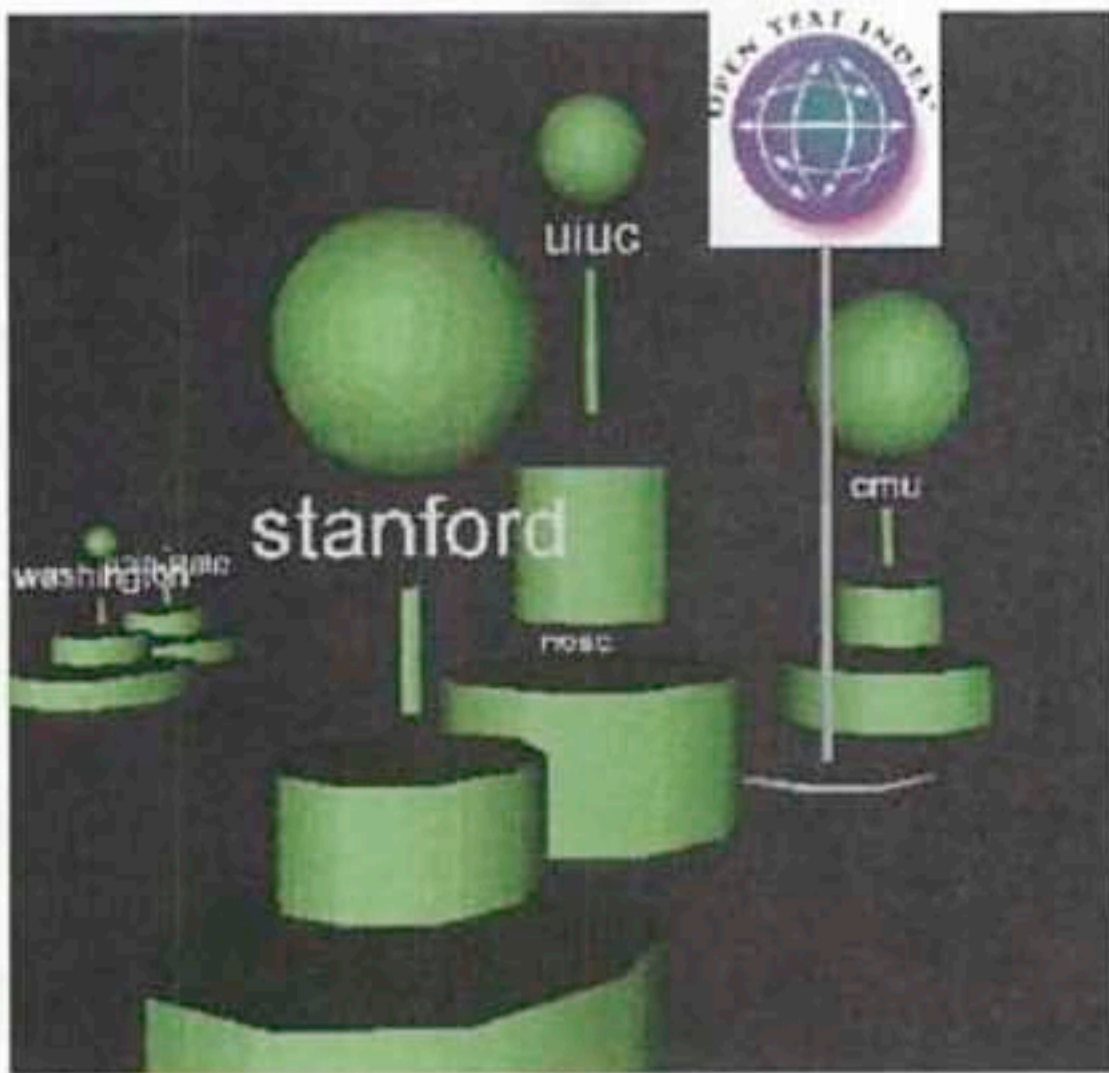
### Visualization Levels of Use

Figure 1.17 also illustrates the application of visualization on at least four levels of use (Card, 1996): (1) visualization of the infosphere, (2) visualization of an information workspace, (3) visual knowledge tools, and (4) visual objects. (See Table 1.2.)

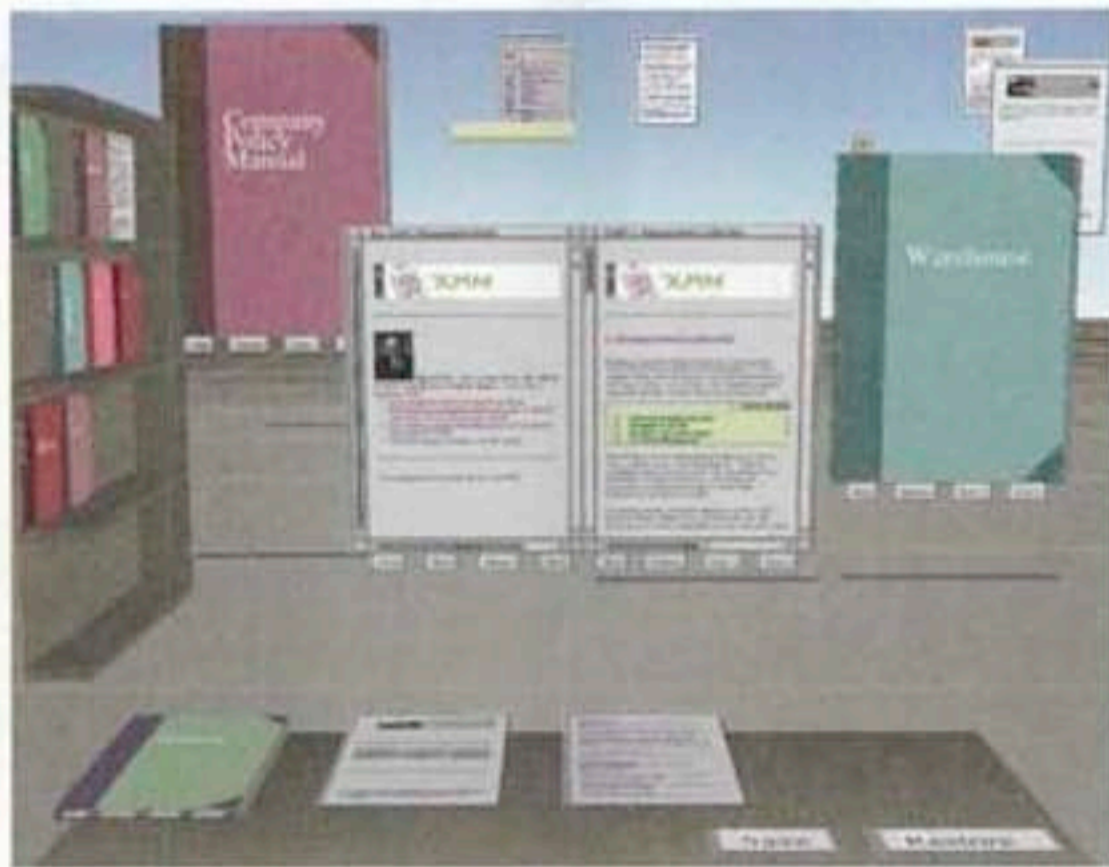
Visualization can be combined with information access techniques to help the user find information. By the *infosphere*, we mean information outside of the user’s work environment. This could be information on the World Wide Web, or it could be information in a specific organizational document collection or digital libraries. The visualization could take the form of a virtual place as in Figure 1.17(a) that contains all the documents, or it could be more abstract.

Visualization of an *information workspace* as shown in Figure 1.17(b)(c) is the use of visualization to organize possibly multiple individual visualizations or other information sources and tools to perform some task. The desktop metaphor for graphical user interfaces (GUIs) performs a similar function. Because information needed is at hand and findable, the time cost of doing some task is reduced, just as a carpentry workbench reduces the time cost of woodworking.

Most visualizations fall at the level of *visual knowledge tools*, as shown in Figure 1.17(d)(e). Either they arrange information to reveal patterns, or they allow the manipulation



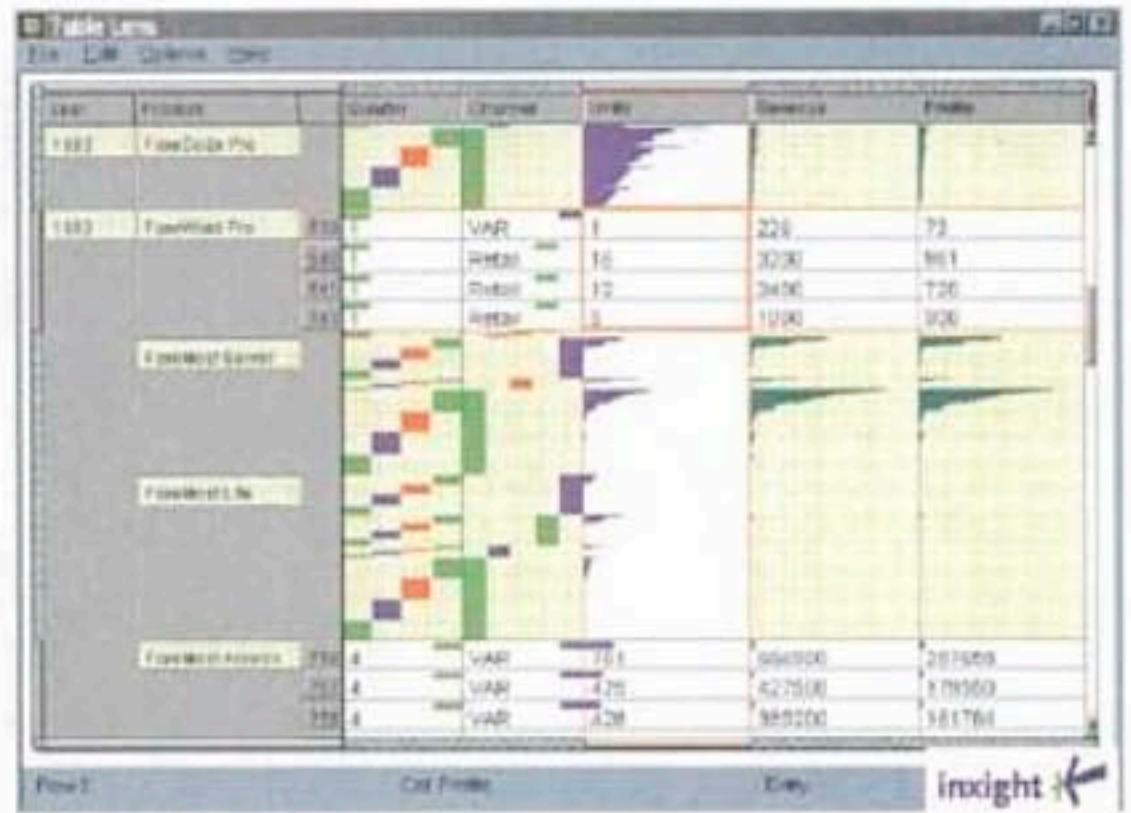
(a) View of sites on the World Wide Web (Bray, 1996 ●, detail from Figure 11).



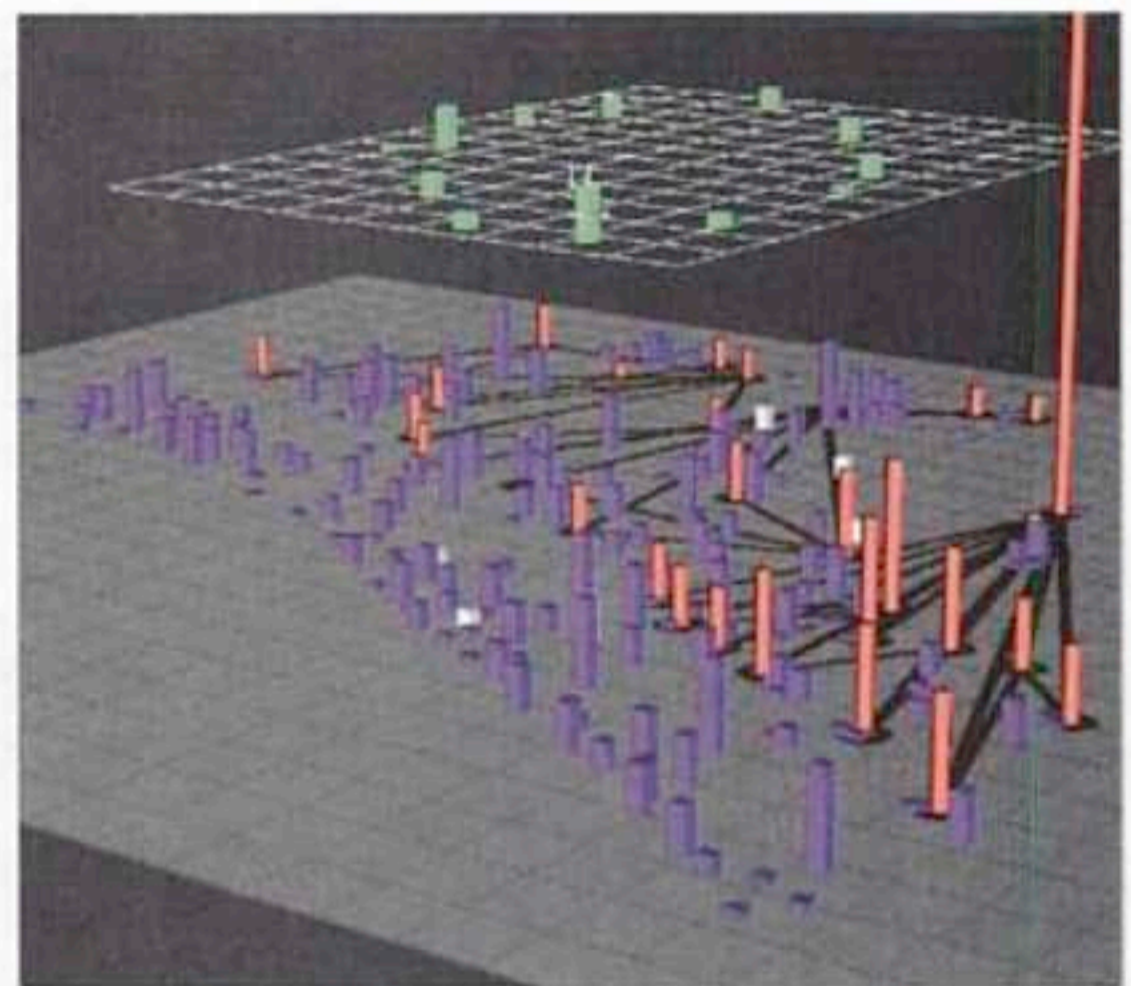
(b) Workspace of Web page. Courtesy of Xerox Corporation. See Card, Robertson, and York (1996 ●).



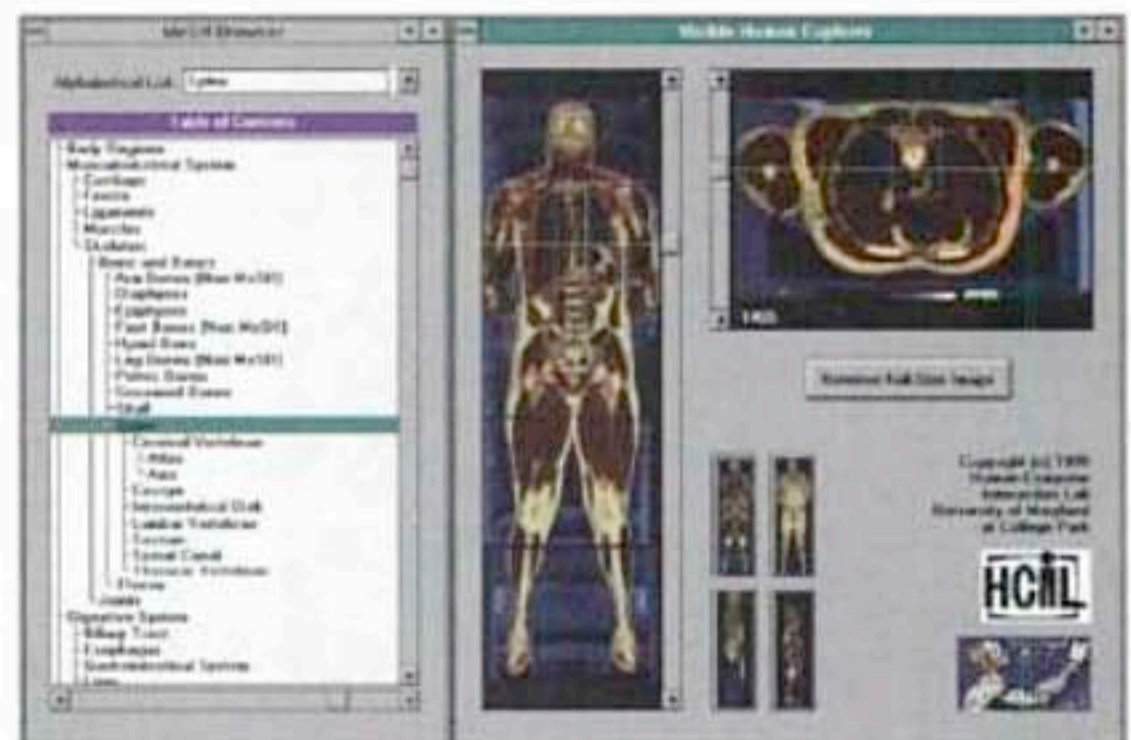
(c) Workspace for document (Risch et al., 1997 ●, detail from Figure 1).



(d) Table Lens tool for data. Courtesy of Inxight Software. See Rao and Card (1994 ●).



(e) SDM tool for logistic data (Chuah et al., 1995a ●).



(f) Human anatomic data packaged as a visualization. Courtesy of the University of Maryland. See North, Shneiderman, and Plaisant (1996 ●).

**FIGURE 1.17**

Examples of information visualization.

TABLE 1.2

Levels at which visualization can be used.

	CONTENTS	EXAMPLE	PRIMARY USE
Infosphere	Information outside the user's environment.	Figure 1.17(a)	Place to find information needed for work.
Information workspace	Information with which the user is interacting as part of some activity.	Figure 1.17(b)(c)	Place to hold work in progress. Used for reducing cost of work, reminding user of work materials.
Visual knowledge tools	A data set.	Figure 1.17(d)(e)	Substrate into which data is poured and/or tool for manipulating it. Used for pattern detection, knowledge crystallization.
Visual objects	One or more data sets packaged for convenience.	Figure 1.17(f)	Packaging of data (data often known in advance). Used to enhance objects of interaction.

of information for finding patterns, or they allow visual calculations. Visual knowledge tools are sometimes called *wide widgets* to emphasize that they are often not just presentations but also controls.

Visualization can also operate at the level of *visually enhanced objects*. These refer to objects, especially virtual physical objects such as the human body or a book, that have been enhanced with visualization techniques to package collections of abstract information. The anatomic browser in Figure 1.17(f), for example, allows both conceptual and spatial browsing of data on a human body.

**Cost Structure**

Figure 1.15 lists some of the principal steps in knowledge crystallization. Each of those actions has a cost associated with it based on the means available for carrying it out. The costs are affected by the representation of information, by the operations available for acting on that information, by various resource capacities affecting the representations and the operations, and by the activity statistics of how often various operations are needed. Together these costs form a *cost structure* of information, a kind of information cost landscape.

Let us illustrate by some examples. Figure 1.18(a) shows a portion of a map of downtown San Francisco. On the



FIGURE 1.18 Cost structure for driving and walking in San Francisco.

map, we have drawn iso-cost contours representing the minimum time to walk to different locations. The operation of walking and the map of San Francisco induce a basic cost structure on the city. In Figure 1.18(b), we have induced a different cost structure by driving. The iso-cost contours are farther apart, since we can go farther for a given amount of cost (in time). Notice also that because there are freeways in the city, the speedup is nonuniform. Representations, defined as data structures + operations + resource constraints, induce different cost structures relative to some task we wish to perform. A rough index of this cost structure is to plot the number of places we could get to for a given cost. That would be a graph with number of places that could be visited increasing approximately as the square of the cost for Figure 1.18(a). The line would be higher for Figure 1.18(b).

The same sort of analysis can apply to the world of information (Card, Pirolli, and Mackinlay, 1994; Card, Robertson, and Mackinlay, 1991; Pirolli and Rao, 1996). Consider, for example, an office worker as shown in Figure 1.19. Information is available in the desk-side diary, through the computer terminal, in the immediate files on the desktop,



FIGURE 1.19 Idealized office layout for optimizing the cost structure of information.

through other people using the telephone, in the books in the bookcase, and in files in the filing cabinet.

The cost structure of the information in the office has been arranged with care. A small amount of information (either frequently needed or in immediate use) is kept where the cost of access is low—in an immediate workspace area, principally the desktop. Voluminous, less used information is kept in a higher-cost, larger-capacity secondary storage area. More information is available in the library and other tertiary storage areas. In addition to these simplified categories, the information is linked and otherwise structured to aid in its retrieval. We could plot the number of documents a user could reach as a function of time (Figure 1.20). We call this diagram a *Cost-of-Knowledge Characteristic Function*. When visualizations are used to help foraging, then the point of a visualization is to raise this curve. If the curve is raised, users can either find the same amount of information in less time or more information in the same amount of time.

The Cost-of-Knowledge Characteristic Function can help us to understand the cost structure of visualizations that aid foraging. Figure 1.21 shows the Spiral Calendar (Mackinlay, Rao, and Card, 1995). In this visualization, calendar representations at different levels of granularity are linked together in such a way that the user can see current information plus information at all higher levels simultaneously. Clicking on a part of a calendar causes that part to expand into a more detailed calendar. The current calendar fragment (and its parents) spiral into the background.

Figure 1.22 shows the Cost-of-Knowledge Characteristic Function for this calendar in comparison to a conventional one on the Sun computer. The comparison is for using only direct point-and-click methods and does not consider string search techniques. The analysis shows that although the Spiral Calendar is superior for very large calendars, the multiple-month technique of conventional calendars results in a lower cost structure for recent dates. The dotted

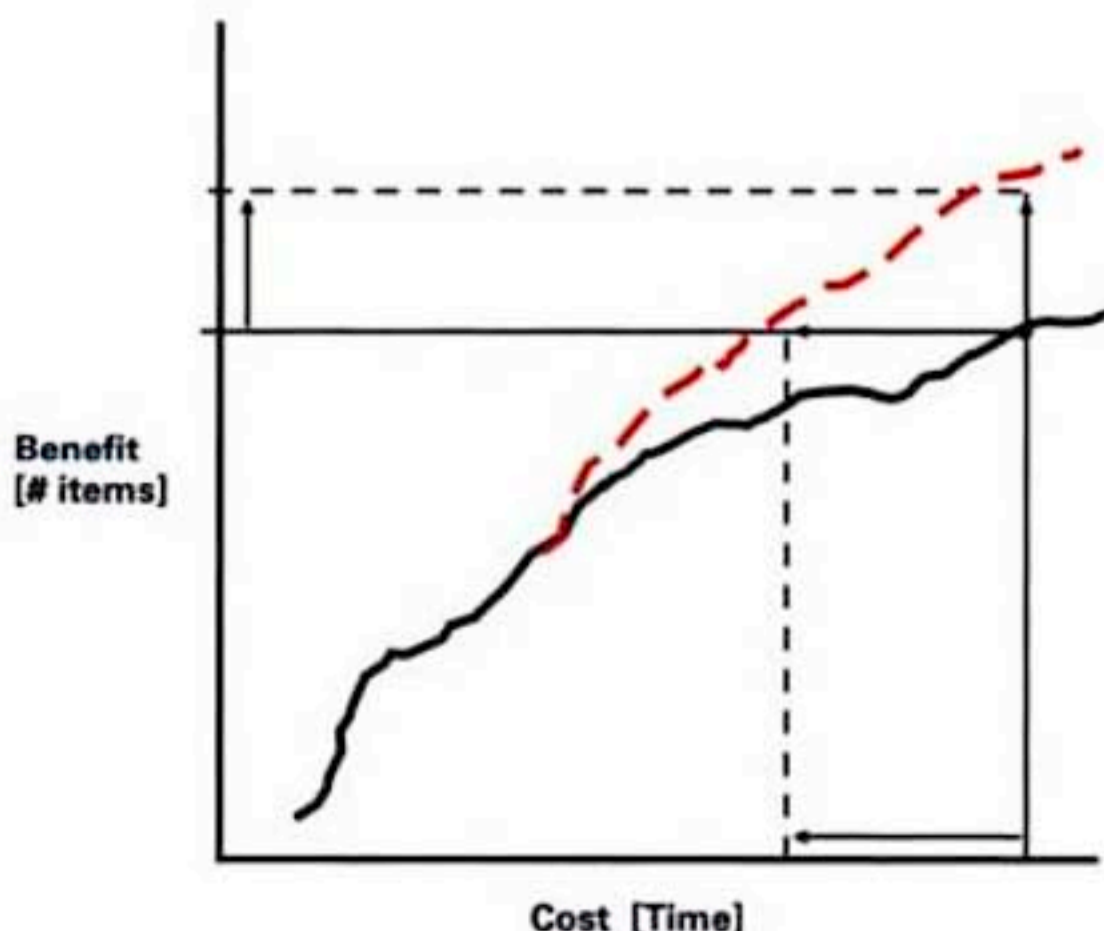


FIGURE 1.20 Cost-of-Knowledge Characteristic Function.

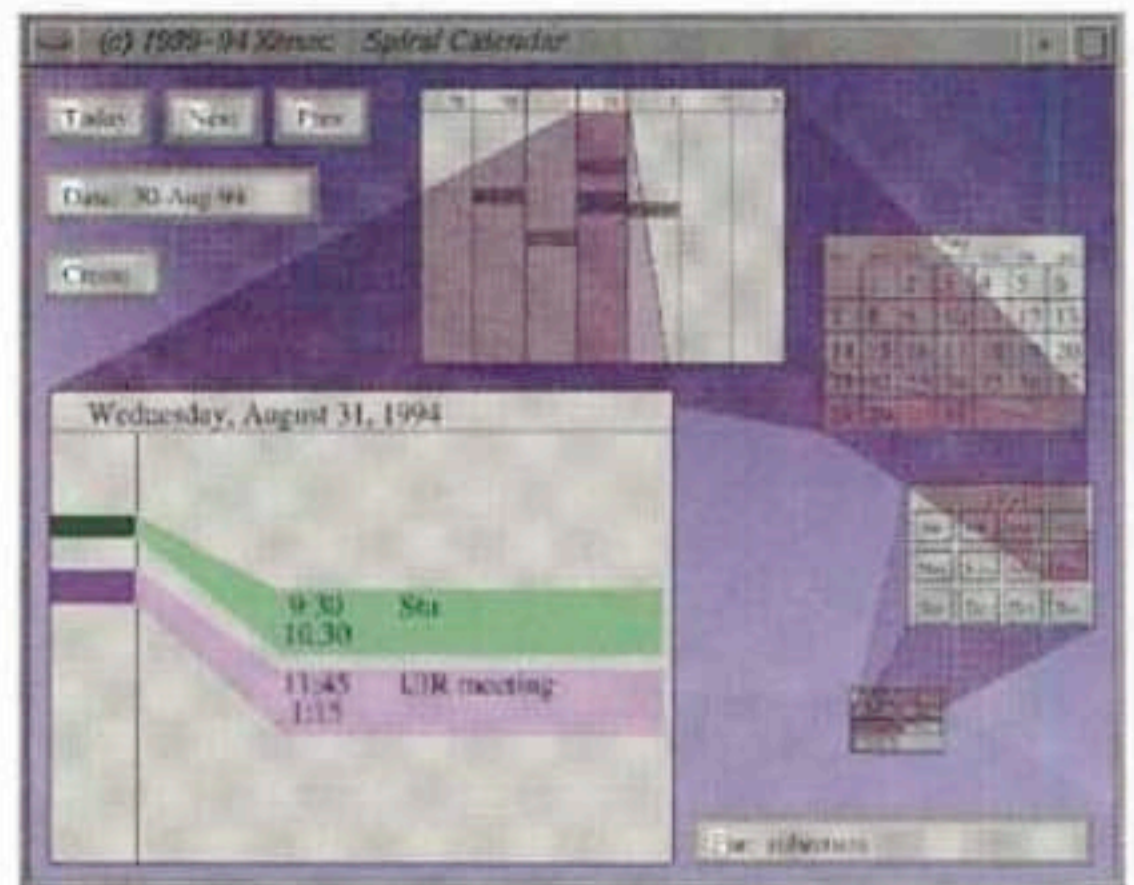


FIGURE 1.21 Spiral Calendar (Card, Pirolli, and Mackinlay, 1994, Figure 2). Courtesy of Xerox Corporation.

lines in the figure are the calculated effects for improvement proposals (some of which were successfully implemented). The Cost-of-Knowledge Characteristic Function is one way to measure the benefits of visualization at least for navigation. The example shows that making effective visualizations is not necessarily easy, even if the visualizations themselves are visually appealing.

### How Visualization Amplifies Cognition

How does visualization amplify cognition? A classic study by Larkin and Simon (1987) illustrates some reasons why visualizations can be effective. Larkin and Simon compared solving physics problems using diagrams versus using non-diagrammatic representations. Specifically, they compared

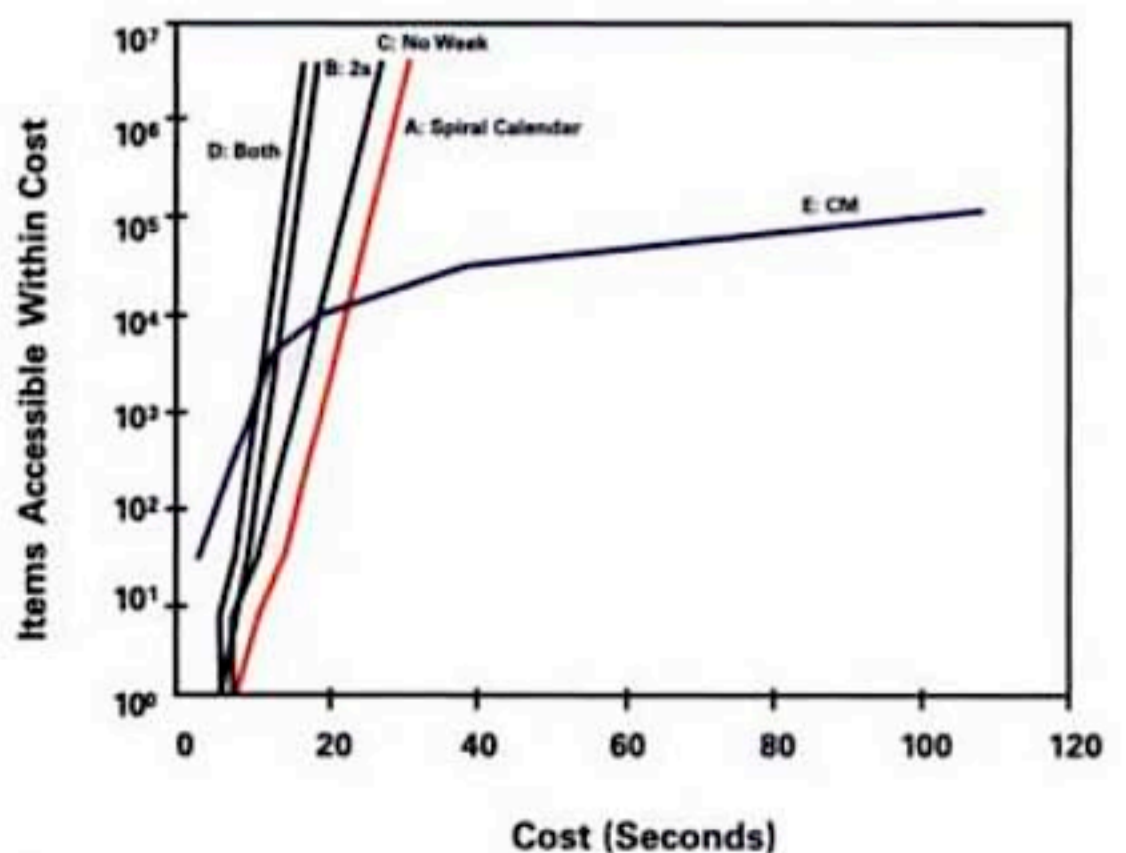


FIGURE 1.22 Cost-of-Knowledge Characteristic Function (Card, Pirolli, and Mackinlay, 1994, Figure 5).

the effort that had to be expended to do search, recognition, and inference with or without the diagram. Their conclusion was that diagrams helped in three basic ways: (1) By grouping together information that is used together, large amounts of search were avoided. (2) By using location to group information about a single element, the need to match symbolic labels was avoided, leading to reductions in search and working memory. (3) In addition, the visual representation automatically supported a large number of perceptual inferences that were extremely easy for humans. For example, with a diagram, geometric elements like alternate interior angles could be immediately and obviously recognized. Two of these ways essentially improve the Cost-of-Knowledge Characteristic Function for accessing information. The third reduces costs of certain operations. The key to understanding the effectiveness of information visualization is understanding what it does to the cost structure of a task. Depending on the task, visualization could make a task better—or it could make the task worse.

We propose six major ways in which visualizations can amplify cognition (Table 1.3): (1) by increasing the memory and processing resources available to the users, (2) by reducing the search for information, (3) by using visual representations to enhance the detection of patterns, (4) by enabling perceptual inference operations, (5) by using perceptual at-

tention mechanisms for monitoring, and (6) by encoding information in a manipulable medium.

Visualizations can expand processing capability by using the resources of the visual system directly. Or they can work indirectly by offloading work from cognition or reducing working memory requirements for a task by allowing the working memory to be external and visual. They can also allow the environment to store details, like a map stores details, close to where they need to be used. As we saw before, if a navigator draws a course on a chart and the course hits a rock, just those depth soundings of most relevance lie near the line he or she has drawn.

Visualizations can reduce the search for data by grouping or visually relating information. They can compact information into a small space. They can allow hierarchical search by using overviews to locate areas for more detailed search. Then they can allow zooming in or popping up details on demand. They can essentially index data spatially by location and landmarks to provide rapid access.

Visualizations can allow patterns in the data to reveal themselves. These patterns suggest schemata at a higher level. Aggregations of data can reveal themselves through clustering or common visual properties.

Visualizations allow some inferences to be done very easily that are not so easy otherwise. This is why all physics

TABLE 1.3

## How information visualization amplifies cognition.

**Increased Resources**

High-bandwidth hierarchical interaction	The human moving gaze system partitions limited channel capacity so that it combines high spatial resolution and wide aperture in sensing visual environments (Resnikoff, 1987).
Parallel perceptual processing	Some attributes of visualizations can be processed in parallel compared to text, which is serial.
Offload work from cognitive to perceptual system	Some cognitive inferences done symbolically can be recoded into inferences done with simple perceptual operations (Larkin and Simon, 1987).
Expanded working memory	Visualizations can expand the working memory available for solving a problem (Norman, 1993).
Expanded storage of information	Visualizations can be used to store massive amounts of information in a quickly accessible form (e.g., maps).

**Reduced Search**

Locality of processing	Visualizations group information used together, reducing search (Larkin and Simon, 1987).
High data density	Visualizations can often represent a large amount of data in a small space (Tufte, 1983).
Spatially indexed addressing	By grouping data about an object, visualizations can avoid symbolic labels (Larkin and Simon, 1987).

**Enhanced Recognition of Patterns**

Recognition instead of recall	Recognizing information generated by a visualization is easier than recalling that information by the user.
Abstraction and aggregation	Visualizations simplify and organize information, supplying higher centers with aggregated forms of information through abstraction and selective omission (Card, Robertson, and Mackinlay, 1991; Resnikoff, 1987).
Visual schemata for organization	Visually organizing data by structural relationships (e.g., by time) enhances patterns.
Value, relationship, trend	Visualizations can be constructed to enhance patterns at all three levels (Bertin, 1977/1981).

**Perceptual Inference**

Visual representations make some problems obvious	Visualizations can support a large number of perceptual inferences that are extremely easy for humans (Larkin and Simon, 1987).
Graphical computations	Visualizations can enable complex specialized graphical computations (Hutchins, 1996).

**Perceptual Monitoring**

Visualizations can allow for the monitoring of a large number of potential events if the display is organized so that these stand out by appearance or motion.

**Manipulable Medium**

Unlike static diagrams, visualizations can allow exploration of a space of parameter values and can amplify user operations.

students are taught to start with a diagram of a problem and high school math students are now taught with graphing calculators. Visual representations can themselves be used for specialized operations.

Thus, as Table 1.3 argues, visualization can enhance cognitive effort by several separate mechanisms. These all depend on appropriate mappings of information into visual form.

## MAPPING DATA TO VISUAL FORM

We can think of visualizations as adjustable mappings from data to visual form to the human perceiver. Figure 1.23 is a diagram of these mappings, to serve as a simple reference model. Using a reference model allows us to simplify our discussion of information visualization systems and to compare and contrast them. Other attempts at reference models are discussed in Robertson and Ferrari (1994).

In Figure 1.23, arrows flow from Raw Data on the left to the human, indicating a series of data transformations. Each arrow might indicate multiple chained transformations. Arrows flow from the human at the right into the transformations themselves, indicating the adjustment of these transformations by user-operated controls. *Data Transformations* map Raw Data, that is, data in some idiosyncratic format, into *Data Tables*, relational descriptions of data extended to include metadata. *Visual Mappings* transform Data Tables into *Visual Structures*, structures that combine spatial substrates, marks, and graphical properties. Finally, *View Transformations* create *Views* of the Visual Structures by specifying graphical parameters such as position, scaling, and clipping. User interaction controls parameters of these transformations, restricting the view to certain data ranges, for example, or changing the nature of the transformation. The visualizations and their controls are used in service of some task.

The core of the reference model is the mapping of a Data Table to a Visual Structure. Data Tables are based on math-

ematical relations; Visual Structures are based on graphical properties effectively processed by human vision. Although Raw Data can be visualized directly, Data Tables are an important intermediate step when the data are abstract, without a direct spatial component. To give an example, text Raw Data might start out as indexed strings or arrays. These might be transformed into document vectors, normalized vectors in a space with dimensionality as large as the number of words. Document vectors might, in turn, be reduced by multidimensional scaling to create Data Tables of  $x, y, z$  coordinates that could be displayed. Whatever the initial form, we assume in our discussion that Raw Data are eventually transformed into the logical equivalent of Data Tables.

The terminology of data in the literature is not consistent (Gallop, 1994; Wong, Crabb, and Bergeron, 1996), since it has been created by many disciplines—mathematics, statistics, engineering, computer science, and graphic design. Consequently, we set out in this section to create a data terminology to be used in the remainder of this book. We have attempted here to strike a balance between formality and clarity (for a more formal treatment see Card and Mackinlay, 1997; Mackinlay, 1986b •; Mackinlay, Card, and Robertson, 1990b). A formal treatment has the virtue that it is precise, which is critical when discussing data, because subtle differences in data often result in large differences in visualization choices. However, clarity is just as important when visualization techniques are being introduced and compared.

### Data Tables

Raw Data comes in many forms, from spreadsheets to the text of novels. The usual strategy is to transform this data into a relation or set of relations that are more structured and thus easier to map to visual forms. Mathematically, a relation is a set of *tuples*:

$$\{ \langle \text{Value}_{ix}, \text{Value}_{iy}, \dots \rangle, \langle \text{Value}_{jx}, \text{Value}_{jy}, \dots \rangle, \dots \}.$$

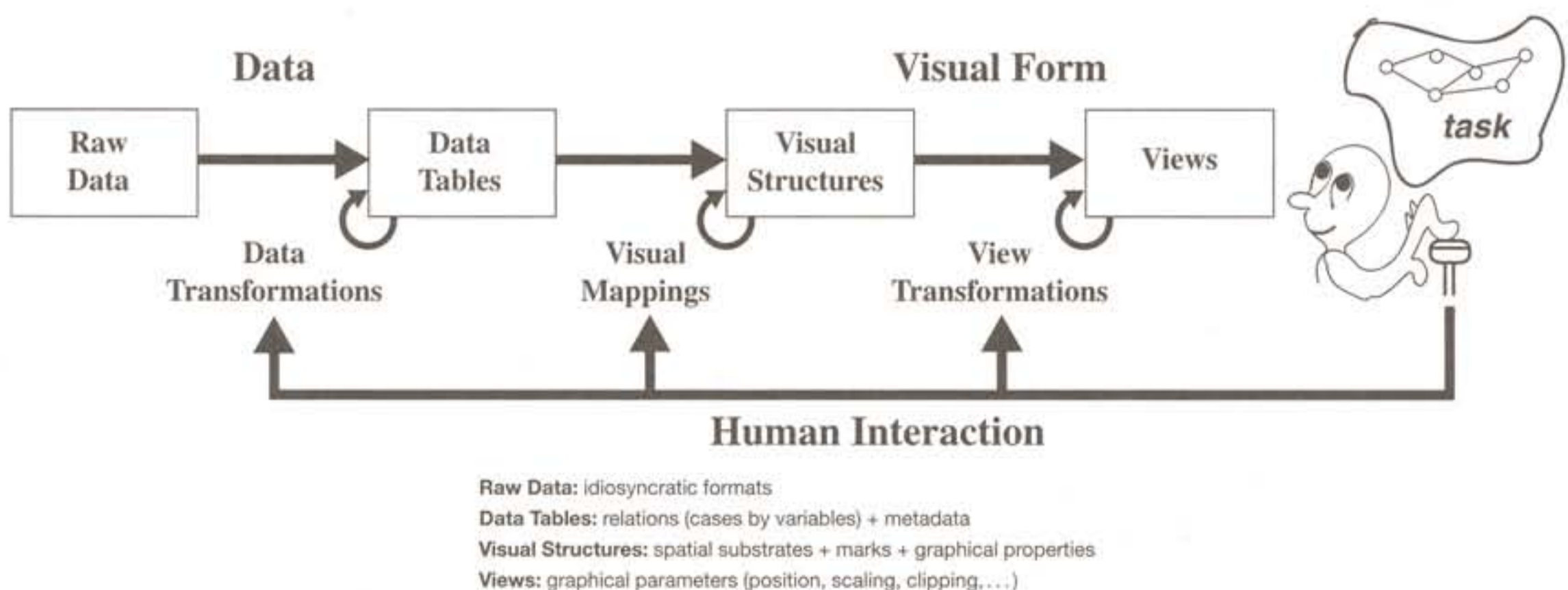


FIGURE 1.23

Reference model for visualization. Visualization can be described as the mapping of data to visual form that supports human interaction in a workspace for visual sense making.



Because this mathematical treatment omits descriptive information that is important for visualization, we create the notion of a *Data Table*. A Data Table (see Table 1.4) combines relations with *metadata* that describes those relations:

TABLE 1.4

A depiction of a Data Table.

	Case <sub>i</sub>	Case <sub>j</sub>	Case <sub>k</sub>	...
Variable <sub>x</sub>	Value <sub>ix</sub>	Value <sub>jx</sub>	Value <sub>kx</sub>	...
Variable <sub>y</sub>	Value <sub>iy</sub>	Value <sub>iy</sub>	Value <sub>ky</sub>	...
...	...	...	...	...

An example of metadata in Table 1.4 are the labels for the rows and columns. The rows represent *variables*, sets that represent the range of the *values* in the tuples. The columns represent *cases*, sets of values for each of the variables. To distinguish a Data Table from other tables (used as presentations of data), we mark Data Tables with a double vertical line on the left of the values. As we shall see, the ordering of the rows and columns in the Data Table may or may not be meaningful. This ordering is another example of metadata that is important for visualization.

Tables of data are often called “cases by variables arrays,” where the cases are the columns in Table 1.4. Cases by variables arrays are often depicted with the cases as rows and the variables as columns, the opposite of our convention here. This is because there are usually many more cases than variables and it is convenient to let the cases expand onto other sheets of paper. On the other hand, when cases are years, as in a budget, the cases are usually laid out as columns. Furthermore, our focus here is on the variables, which are important when selecting visualizations (the cases are important when analyzing data). Therefore, for expository convenience (large numbers of cases are not necessary in examples), we have chosen to depict Data Tables with the cases as columns and variables as rows. Bertin (1977/1981) also follows this Data Table convention and depicts the cases as columns and the variables as rows, but he calls the cases “objects” and the variables “characteristics.” His terminology, however, focuses on a specialized form of relation called a *function*, which has the mathematical property that variables are divided into *inputs* and *outputs* and the input variables uniquely determine the output variables. Functions from objects to their characteristics are very common in the tasks associated with visualization. They have one input variable and an arbitrary number of output variables, where each case represents a unique object:

$$f(\text{Case}_i) = \langle \text{Value}_{ix}, \text{Value}_{iy}, \dots \rangle.$$

We depict functions in Data Tables by separating the input variables from the output variables with a thick line as shown in Table 1.5. In this table, since *Case* is a variable in the Data Table, it is no longer metadata.

TABLE 1.5

A function described in a Data Table with input variables shown above the output variables. Case<sub>i</sub> represents a unique object and the corresponding values represent the characteristics of that object.

Case	Case <sub>i</sub>	Case <sub>j</sub>	Case <sub>k</sub>	...
Variable <sub>x</sub>	Value <sub>ix</sub>	Value <sub>jx</sub>	Value <sub>kx</sub>	...
Variable <sub>y</sub>	Value <sub>iy</sub>	Value <sub>iy</sub>	Value <sub>ky</sub>	...
...	...	...	...	...

One of the advantages of Data Tables is that they clearly depict the number of variables associated with a collection of data, an important consideration when selecting visualizations. “Dimensionality” is one of those terms used in different ways by different authors (Wong, Crabb, and Bergeron, 1996). Dimensionality is used to refer to the number of input variables, the number of output variables, both together, or even the number of spatial dimensions in the data. The term is also commonly used to describe the type of spatial substrate of a Visual Structure. The dimensionality of space, whether it describes data or Visual Structures, is the most popular use of this term and how we generally use it in this book. Two-dimensional Visual Structures are the largest we can visualize before we have to worry about occlusions, for although we live in a 3D world, our vision (unless we move) sees something like the inside surface of a 2D sphere. Three-dimensional Visual Structures are the largest we can access with our specialized human perceptual operations. We follow common usage of the term “multi” and apply *multivariable* to data (as opposed to visualizations), specifically to Data Tables that have too many variables to be encoded in a single 3D Visual Structure. Visualizations that are specifically designed to encode such multivariable Data Tables are called *multidimensional* visualizations.

Now that we have established some data terminology, we can use Data Tables to clarify some issues associated with visualizing data. Table 1.6 describes a Data Table for films where the cases (columns) represent films and the variables (rows) represent properties of those films:

TABLE 1.6

A Data Table about films.

<i>FilmID</i>	230	105	540	...
<i>Title</i>	Goldfinger	Ben Hur	Ben Hur	...
<i>Director</i>	Hamilton	Wyer	Niblo	...
<i>Actor</i>	Connery	Heston	Novarro	...
<i>Actress</i>	Blackman	Harareet	McAvoy	...
<i>Year</i>	1964	1959	1926	...
<i>Length</i>	112	212	133	...
<i>Popularity</i>	7.7	8.2	7.4	...
<i>Rating</i>	PG	G	G	...
<i>Film Type</i>	Action	Action	Drama	...

This table could have been written without any input variables, but we have included one, *FilmID*, which is a set of unique numbers identifying the films. The other properties (for example, *Title*) do not have unique values for each case. Such identifiers or codes are often maintained as a key by relational databases when there is no other key for a record. Because it is unique for a case, *FilmID* can be used to index a mapping from films to marks on a spatial substrate that encodes them.

Most tables used to present data are not Data Tables. Take Table 1.7, a Data Table that describes distances between cities:

TABLE 1.7

Data Table for distances.

<i>Start City</i>	Basel	Basel	Berlin	...
<i>End City</i>	Berlin	Bern	Bern	...
<i>Distance</i>	860	90	930	...

Table 1.7 is an example of a function with two input variables. Such data is often presented as a two-way table (Table 1.8). Table 1.7 is a Data Table, whereas Table 1.8 is not. It is an instance of a table presentation.

TABLE 1.8

A table presentation for the same distances. This is not a Data Table.

	<b>Basel</b>	<b>Berlin</b>	<b>Bern</b>	...
<b>Basel</b>	0	860	90	...
<b>Berlin</b>	860	0	930	...
<b>Bern</b>	90	930	0	...
...	...	...	...	...

Table 1.8 is effective for seeing the distances between cities. Considered as a presentation, Table 1.7 is effective for seeing the structure of the data.

Data Tables can undergo data transformations that affect their structure. For example, Table 1.7 could have been derived by a data transformation from Table 1.9.

TABLE 1.9

Possible earlier form of Data Table 1.7.

<i>City</i>	Basel	Berlin	Bern	...
<i>Latitude</i>	47.33N	52.32N	46.57N	...
<i>Longitude</i>	7.36E	13.25E	7.26E	...
<i>Country</i>	SWTZ	GER	SWTZ	...
...	...	...	...	...

In Table 1.9, the input variable *City* is mapped to various output variables, including *Latitude* and *Longitude*, which can be used to calculate the *Distance* variable in Data Table 1.7. Thus, the transformation from Data Table 1.9 to Data Table 1.7 involves both new *derived values* and new *derived structure*. It involves new derived values because the *Distance* values have been computed from other values. It involves new derived structure because the numbers and identities of input and/or output variables have changed between the two Data Tables. In fact, some output variables have been used to create a new input variable. Such Data Table transformations are common as data are mapped to visual form.

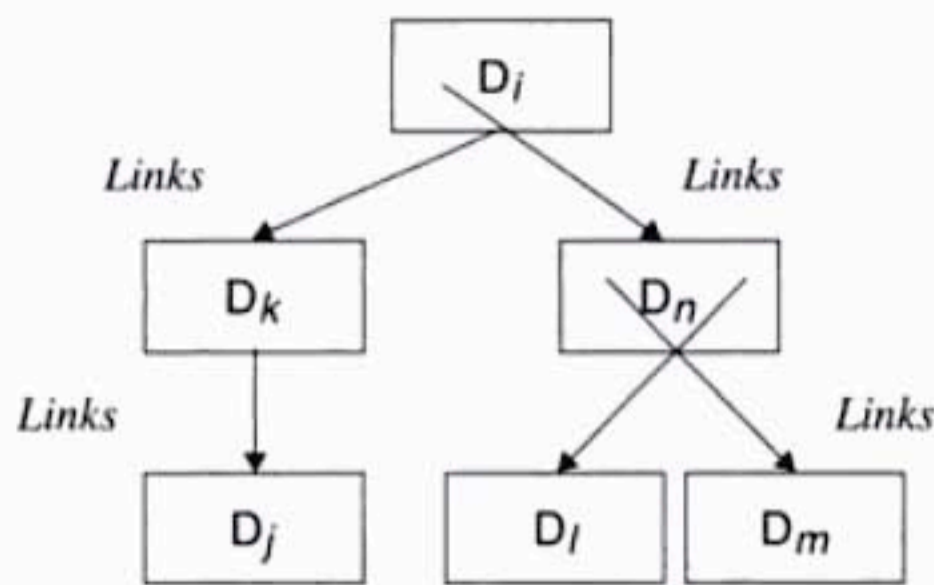
Data Tables can describe hierarchical and network data. To do this, a variable is used to describe the links between cases. For example, in Table 1.10 the variable *Links* describes the relationship among hypertext documents:

TABLE 1.10

Data Table describing the links among hypertext documents.

<i>DocID</i>	$D_j$	$D_l$	$D_k$	$D_l$	$D_m$	$D_n$
<i>Length</i>	235	54	127	341	102	186
<i>Links</i>	$\{D_k, D_n\}$	$\emptyset$	$\{D_l\}$	$\emptyset$	$\emptyset$	$\{D_k, D_m\}$
...	...	...	...	...	...	...

These links form the following hierarchy:



Hierarchies are specialized networks with one root and with each child having exactly one parent. Notice that the values of *Links* are sets that contain the *DocIDs* of the cases (or the null set  $\emptyset$ ) and that this variable represents a mapping from a set of cases back into itself. This self-referential property of *Links* is included in the metadata associated with Data Table 1.10.

**Variable Types**

Variables come in three basic types:

- N = *Nominal* (are only = or  $\neq$  to other values),
- O = *Ordinal* (obeys a < relation), or
- Q = *Quantitative* (can do arithmetic on them).

A *nominal variable* N is an unordered set, such as film titles {Goldfinger, Ben Hur, Star Wars}. An *ordinal variable* O is a tuple (ordered set), such as film ratings <G, PG, PG-13, R>. A *quantitative variable* Q is a numeric range, such as film length [0, 360]. These distinctions are important, because they determine the type of axis that should be used in a Visual Structure.

Elementary choices for data transformations derive from the variables types. For example, quantitative variables can be transformed into ordinal variables

$$Q \rightarrow O$$

by dividing them into ranges. Film lengths (type Q)

$$[0, 360]$$

can be broken into the ranges (type O)

$$\langle \text{Short, Medium, Long} \rangle.$$

This common transformation is called *classing*, because it maps values onto classes of values. It creates an accessible summary of the data, although it loses information. A more sophisticated variation creates an additional variable that counts the values in the ranges, leading to a histogram. A less common transformation converts ordinal variables into nominal variables  $O \rightarrow N$  by ignoring the ordering. In the other direction, nominal variables can be sorted to create ordinal variables

$$N \rightarrow O.$$

For example, film titles

$$\{ \text{Goldfinger, Ben Hur, Star Wars} \}$$

can be sorted lexicographically

$$\langle \text{Ben Hur, Goldfinger, Star Wars} \rangle.$$

In addition to the three basic types of variables, there are subtypes that represent important properties of the world associated with specialized visual conventions. We distinguish the subtype

$$Q_s = \text{Quantitative Spatial}$$

for intrinsically spatial variables common in scientific visualization, and the subtype

$$Q_g = \text{Quantitative Geographical}$$

for spatial variables that are specifically geophysical coordinates.

Other important subtypes are the temporal variables

$$Q_t = \text{Quantitative Time}$$

and

$$O_t = \text{Ordinal Time}.$$

Temporal variables have associated data transformations, such as collecting days into weeks, months, or years. Of course, natural numbers, used as counting numbers, are another important subtype.

**Metadata**

*Metadata* is descriptive information about data (see Tweedie, 1997 •). Metadata can be important in choosing visualizations. For example, Table 1.11 (Gallop, 1994) describes a function from map locations to numbers.

TABLE 1.11

Data Table for map numbers.

Latitude	$Y_i$	$Y_j$	$Y_k$	...
Longitude	$X_i$	$X_j$	$X_k$	...
Numbers	$Q_i$	$Q_j$	$Q_k$	...

If the *Numbers* variable represents height above sea level, the relation represents samples from a continuous real function, which can be interpolated to approximate a surface. On the other hand, if *Numbers* represents car accidents, that is to say, natural numbers, it is not permissible to interpolate.

An important form of metadata is the *structure* of a Data Table (Tweedie, 1997 •), which is depicted as the rows and columns in our Data Table examples. Data transformations often change the structure of a Data Table. A document's location in a semantic space could be represented using three variables X, Y, and Z or described by a single vector variable *Location*. A group of survey respondents could be individual cases described by output variables *Age* and *Sex*, or

alternately the group could be classed into "cases"  $Age < 20$ ,  $Age 20-35$ ,  $Age > 35$  with  $Age$  and  $Sex$  as input variables whose values were sets of respondent identifier codes.

Additional metadata could be added explicitly to the Data Table by adding, for example, a column for data type as in Table 1.12.

TABLE 1.12

A Data Table with metadata describing the types of the variables.

<i>FilmID</i>	N	230	105	...
<i>Title</i>	N	Goldfinger	Ben Hur	...
<i>Director</i>	N	Hamilton	Wyler	...
<i>Actor</i>	N	Connery	Heston	...
<i>Actress</i>	N	Blackman	Harareet	...
<i>Year</i>	$Q_t$	1964	1959	...
<i>Length</i>	Q	112	212	...
<i>Popularity</i>	Q	7.7	8.2	...
<i>Rating</i>	O	PG	G	...
<i>Film Type</i>	N	Action	Action	...

Additional columns could be added for cardinality or range of the data. Data Tables can also include relationships between variables that are not easily depicted. For example, a business database may contain two relations: employees and sales. The sales relation will have a variable for the person who made the sale, which will be a subset of an employees variable.

**Data Transformations**

The transformation of Raw Data into Data Tables typically involves the loss or gain of information. Often Raw Data contains errors or missing values that must be addressed before the data can be visualized. Statistical calculations can also add additional information. For these reasons, Data Tables often contain derived value or structure. There are four types of these data transformations (Tweedie, 1997 ●):

1. Values → Derived Values
2. Structure → Derived Structure
3. Values → Derived Structure
4. Structure → Derived Values

Examples of these occur in Table 1.13.

TABLE 1.13

Examples of data transformations.

	Derived Value	Derived Structure
Value	<i>Mean</i>	<i>Sort</i> <i>Class</i> <i>Promote</i>
Structure	<i>Demote</i>	$X, Y, Z \rightarrow P_{xyz}$

Statistical calculations, like *Mean*, are an example of derived values. Sorting variables or cases is an example of derived structure (Bertin, 1977/1981).

Transformations that switch between value and structure are more complex. Data transformations can be concatenated to form chains of aggregation and classing as part of the knowledge crystallization process shown in Figure 1.15. Patterns can be discovered and brought forward as new schemata by encoding them in the variables of the Data Table. Visualizations of the Data Table can be used to detect more patterns. User-operated controls on structural transformations of the Data Table can be used as controls on the visualization. An example of chained value and structure transformations is the "aggregation cycle" described by Bertin (1977/1981): Data Table 1.14 describes individuals and their ages, income, and profession:

TABLE 1.14

A Data Table describing individuals and their ages, incomes, and professions.

<i>Individual</i>	I1	I2	I3	I4	I5	I6	I7	I8	...
<i>Ages</i>	55	18	22	51	34	50	28	17	...
<i>Income</i>	1	6	8	10	4	7	3	1	...
<i>P1</i>	0	0	0	0	1	0	0	0	...
<i>P2</i>	1	1	0	0	0	0	0	0	...
<i>P3</i>	0	0	0	0	0	0	0	0	...
<i>P4</i>	0	0	1	0	0	0	1	0	...
<i>P5</i>	0	0	0	0	0	0	0	1	...
<i>P6</i>	0	0	0	1	0	1	0	0	...
<i>P7</i>	0	0	0	0	0	0	0	0	...
<i>P8</i>	0	0	0	0	0	0	0	0	...

Ages and Income are quantitative variables. Variables P1 through P8 represent different professions, with a "1" value indicating that individual has that profession.

The first step in the aggregation cycle is to transform the quantitative variables of Ages and Income into ordinal variables of age classes and income classes, creating the Data Table 1.15 consisting entirely of binary data values:

Class (Table 1.14) on Ages and Income → Table 1.15,

where, to keep the example simple we omit specification of the obvious parameters for specifying class boundaries, scope of aggregation, and so on.

TABLE 1.15

The age and income classes derived from Table 1.14.

Individual	I1	I2	I3	I4	I5	I6	I7	I8	...
Age>40	1	0	0	1	0	1	0	0	...
Age20-40	0	0	1	0	1	0	1	0	...
Age0-20	0	1	0	0	0	0	0	1	...
Inc7-10	0	0	1	1	0	1	0	0	...
Inc4-6	0	1	0	0	1	0	0	0	...
Inc2-3	0	0	0	0	0	0	1	0	...
Inc0-1	1	0	0	0	0	0	0	1	...
P1	0	0	0	0	1	0	0	0	...
P2	1	1	0	0	0	0	0	0	...
P3	0	0	0	0	0	0	0	0	...
P4	0	0	1	0	0	0	1	0	...
P5	0	0	0	0	0	0	0	1	...
P6	0	0	0	1	0	1	0	0	...
P7	0	0	0	0	0	0	0	0	...
P8	0	0	0	0	0	0	0	0	...

This transformation involves Structure → Derived Structure with the creation of the new variables for the ranges, whose rows are ordered. It also involves Values → Derived Values with the calculation of the binary values for each individual to indicate their age and income ranges.

We next generate the new Table 1.16 by aggregating individuals into their professional groups. The professions become the cases and the number of individuals in each age and income class become the new Data Values. We call this operation promotion, meaning that a variable is promoted into being a case (i.e., the level of the case has been promoted to a higher level of aggregation):

Promote (Table 1.15) on Professions classes → Table 1.16.

TABLE 1.16

Promotion of professions to cases.

P-ID	P1	P2	P3	P4	P5	P6	P7	P8
Age>40	0	0	1	0	2	2	1	1
Age20-40	3	1	0	0	0	0	0	2
Age0-20	0	2	1	1	3	1	0	1
Inc7-10	1	0	0	1	2	2	0	1
Inc4-6	2	2	0	0	1	0	0	2
Inc2-3	0	1	1	2	0	1	1	1
Inc0-1	0	1	0	3	1	0	0	0

This transformation involves Structure → Derived Values when the professions become the values for a new input variable.

A new cycle can start from Data Table 1.16 by calculating the mean Age and Income of each profession:

Mean (Table 1.16) on Age and Income → Table 1.17

TABLE 1.17

Average age and income of the professions.

P-ID	P1	P2	P3	P4	P5	P6	P7	P8
Avg Age	33	29	17	34	25	40	58	31
Avg Income	6.3	3.7	3	2.7	3.5	6.6	2	5.7

Again, this is a Values → Derived Values.

These quantitative variables can then be transformed to ordinal variables representing classes of median age and income:

Class (Table 1.17) on AveAge and AveIncome → Table 1.18.

TABLE 1.18

Classing of average age and income.

P-ID	P1	P2	P3	P4	P5	P6	P7	P8
Avg Age>35	0	0	0	0	0	1	1	0
Avg Age20-35	1	1	0	1	1	0	0	1
Avg Age0-20	0	0	1	0	0	0	0	0
Avg Inc>6	1	0	0	0	0	1	0	0
Avg Inc5-6	0	0	0	0	0	0	0	1
Avg Inc4-5	0	0	0	0	0	0	0	0
Avg Inc3-4	0	1	1	0	1	0	0	0
Avg Inc<3	0	0	0	1	0	0	1	0

This is another *Structure* → *Derived Structure* transformation.

We can then treat average income as a case (Bertin calls these statistical objects) resulting in a cross-tabulation table:

Promote (Table 1.18) on AveAge and AveInc classes → Table 1.19

TABLE 1.19

Promotion of average income classes.

Avg Inc-ID	AI>6	AI5-6	AI4-5	AI3-4	AI<3
Avg Age>35	1	0	0	0	1
Avg Age20-35	1	1	0	2	1
Avg Age0-20	0	0	0	1	0

This cycle can be continued. The example, summarized in Figure 1.24, also illustrates the complexities of data transformation and the kinds of transformations we would like to be able to visualize and maybe control through visualizations. Each of these Data Tables reveals a different aspect of the data and may lead to a different choice of Visual Structure. We return to the problem of choosing visualizations after discussing Visual Structures and View.

### VISUAL STRUCTURES

In visualization, Data Tables are mapped to *Visual Structures*, which augment a spatial substrate with marks and graphical properties to encode information. To be a good Visual Structure, it is important that this mapping preserve the data (Mackinlay, 1986b •). Data Tables can often be mapped into the visual representations in multiple ways. A mapping is said to be *expressive* if all and only the data in the Data Table are also represented in the Visual Structure. Good mappings are difficult, because it is easy for unwanted data to appear in the Visual Structure. For example, the visual presentation in Figure 1.25 is not expressive. It uses an ordinal axis in the Visual Structure to express a nominal relationship in the Data Table. It expresses visually a relationship not in the data.

The mapping must also be one that can be perceived well by the human. A mapping is said to be more *effective* if it is faster to interpret, can convey more distinctions, or leads to fewer errors than some other mapping. In Figure 1.26, the mapping of the sine wave into position is more effective than the mapping into color.

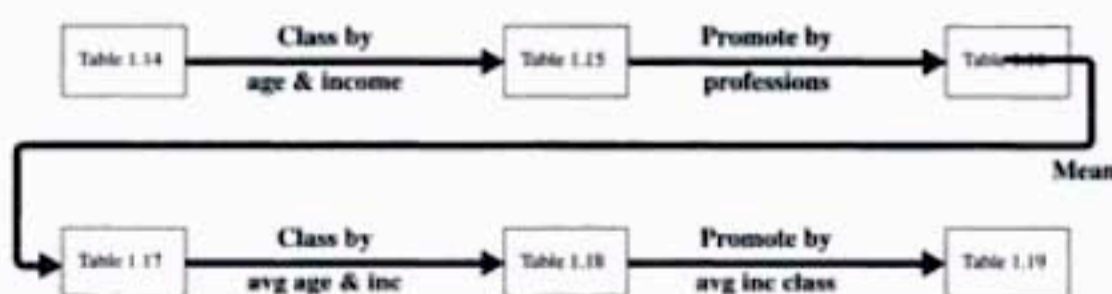
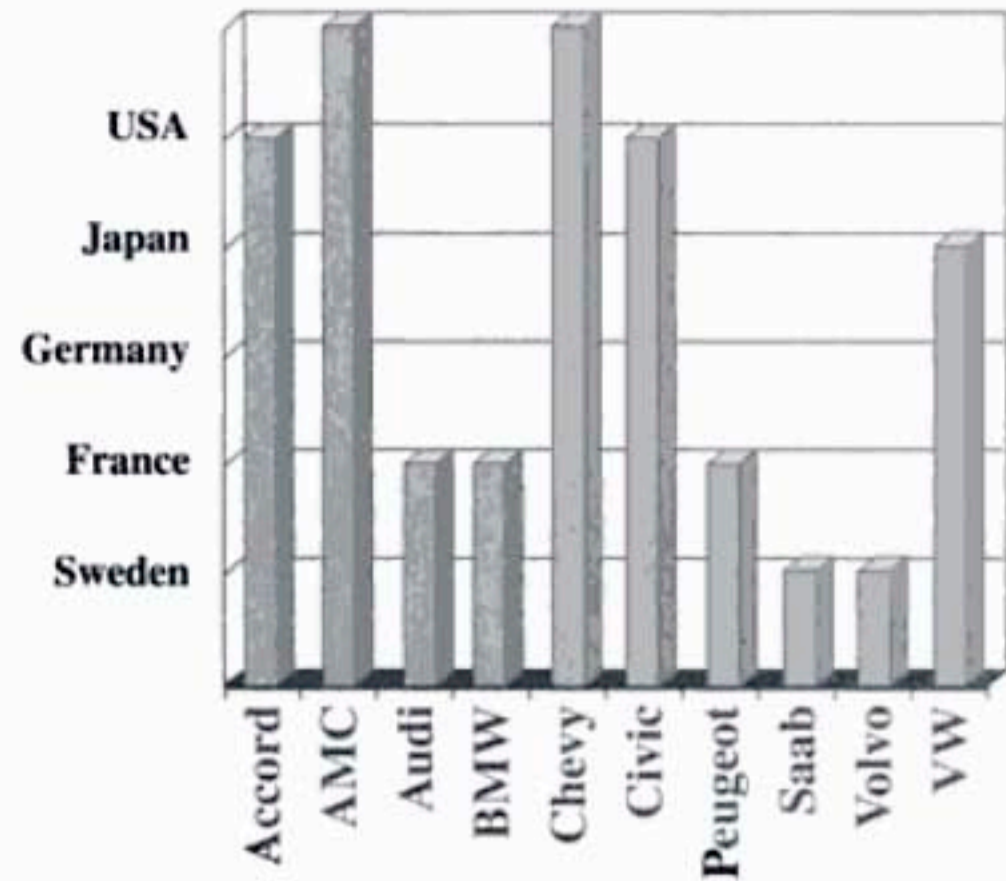


FIGURE 1.24

Cascaded data transformations.



Incorrect!

FIGURE 1.25

This Visual Structure is not expressive, because it implies incorrect ordinal relationship among countries.

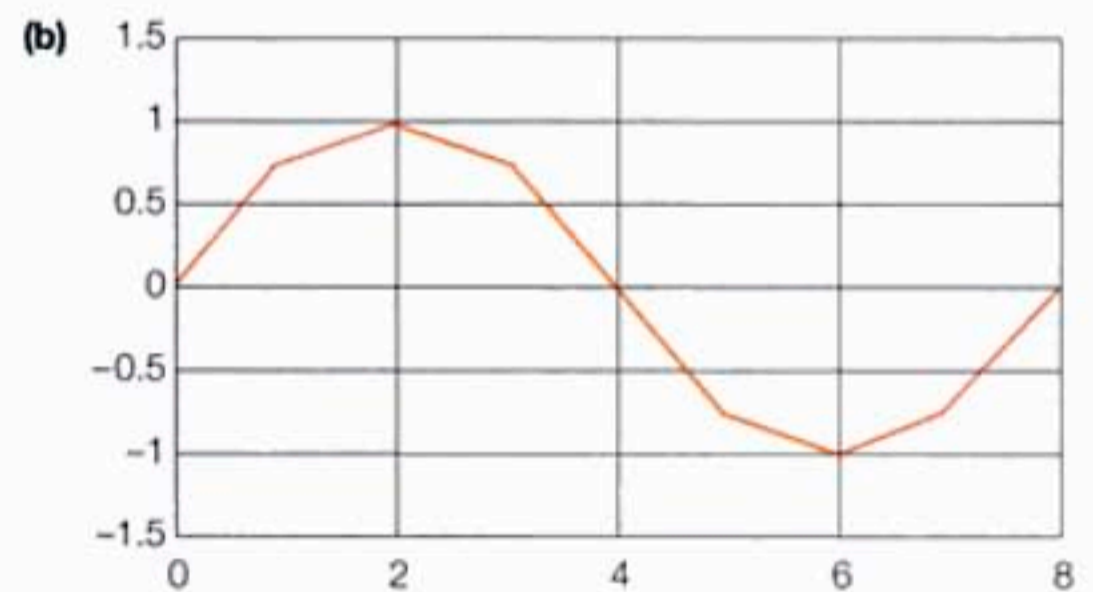


FIGURE 1.26

Effectiveness of visual representations. (a) is less effective than (b) for communicating a sine wave.

To understand effectiveness, we have to understand a few rudimentary facts from perception. One set of such facts concern perceptual characteristics of the different graphical representations. But another set of facts concern the way in which perception itself is an active system of shifting attention, a characteristic we can attempt to play to in information visualizations.

### Perception

Information visualization is clearly dependent upon the properties of human perception. Perception is a vast and

studied subject (see, for example, Atkinson et al., 1988; Boff, Kaufman, and Thomas, 1986; Kosslyn, 1994; Tovée, 1996). Until recently, however, the connection between perception and cognitive activities has been tenuous (Elkind et al., 1990), making external cognition (such as the tasks of information visualization) difficult to study with any precision. While summarizing the literature of perception and addressing the integration of perceptual and cognitive theories are clearly beyond the scope of this book, we can give here a few selected facts about perception that are useful for visualization.

It is the job of information visualization systems to set up visual representations of data so as to bring the properties of human perception to bear. At the most basic level, the visual perceptual system uses a three-level hierarchical organization to partition limited bandwidth between the conflicting needs for both high spatial resolution and wide aperture in sensing the visual environment (Resnikoff, 1987). It is possible to exploit this organization in designing visualizations.

Figure 1.27 shows the human eye. A movable lens is imaged onto a substrate of 125 million photoreceptors, comprising 6.5 million color-detecting cones and the rest black and white detecting rods. Distribution of these photoreceptors is nonuniform (Figure 1.28). In a central area, called the *fovea*, cones are dense. In outlying areas, rods with larger receptive fields predominate.

Figure 1.29 shows a logical map of the eye. The first level of the visual system (see Resnikoff, 1987) is the retina. The retina has an area of about  $1000 \text{ mm}^2 = 10^9 \mu\text{m}^2$  and covers a visual field of about  $160^\circ$  wide (since the two eyes are set horizontally and their visual fields only partly overlap, together they cover a visual field at the extremes roughly  $200^\circ$  horizontally and  $135^\circ$  vertically). The density of cones in the nonfoveal portions of the retina is about  $0.006 \text{ cones}/\mu\text{m}^2$ . The organization of this part of the retina is good at detecting

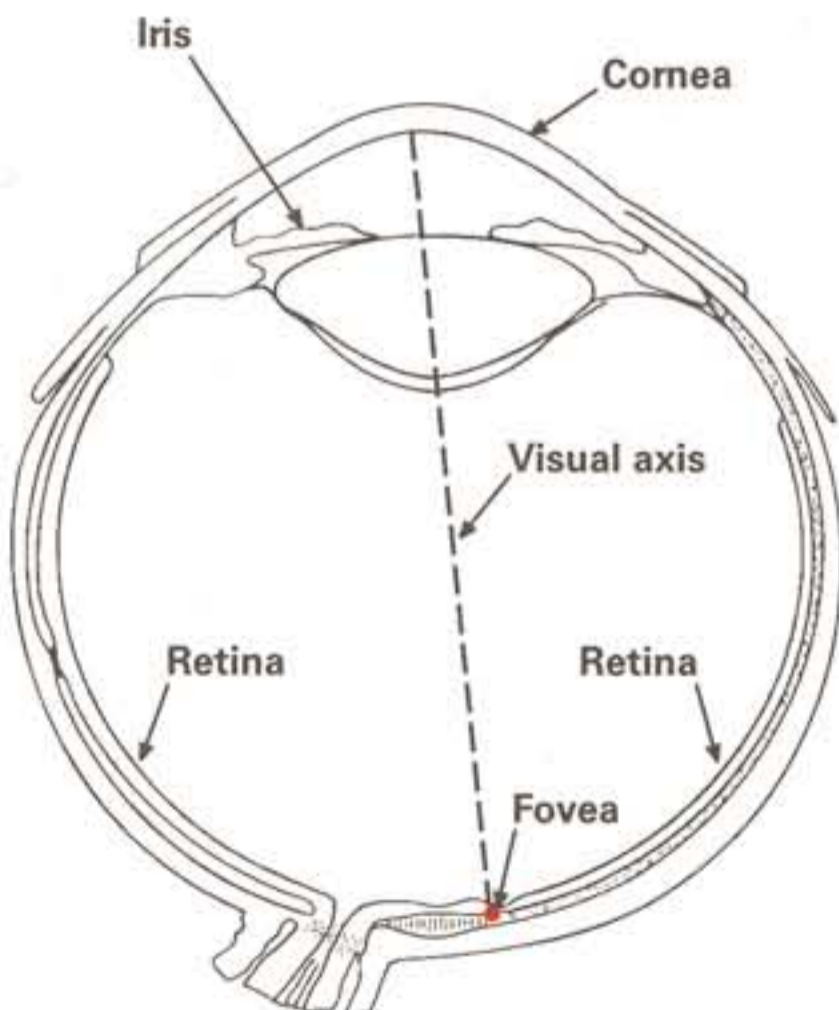


FIGURE 1.27 The human eye. By permission of Resnikoff (1987, Figure 5.3.2).

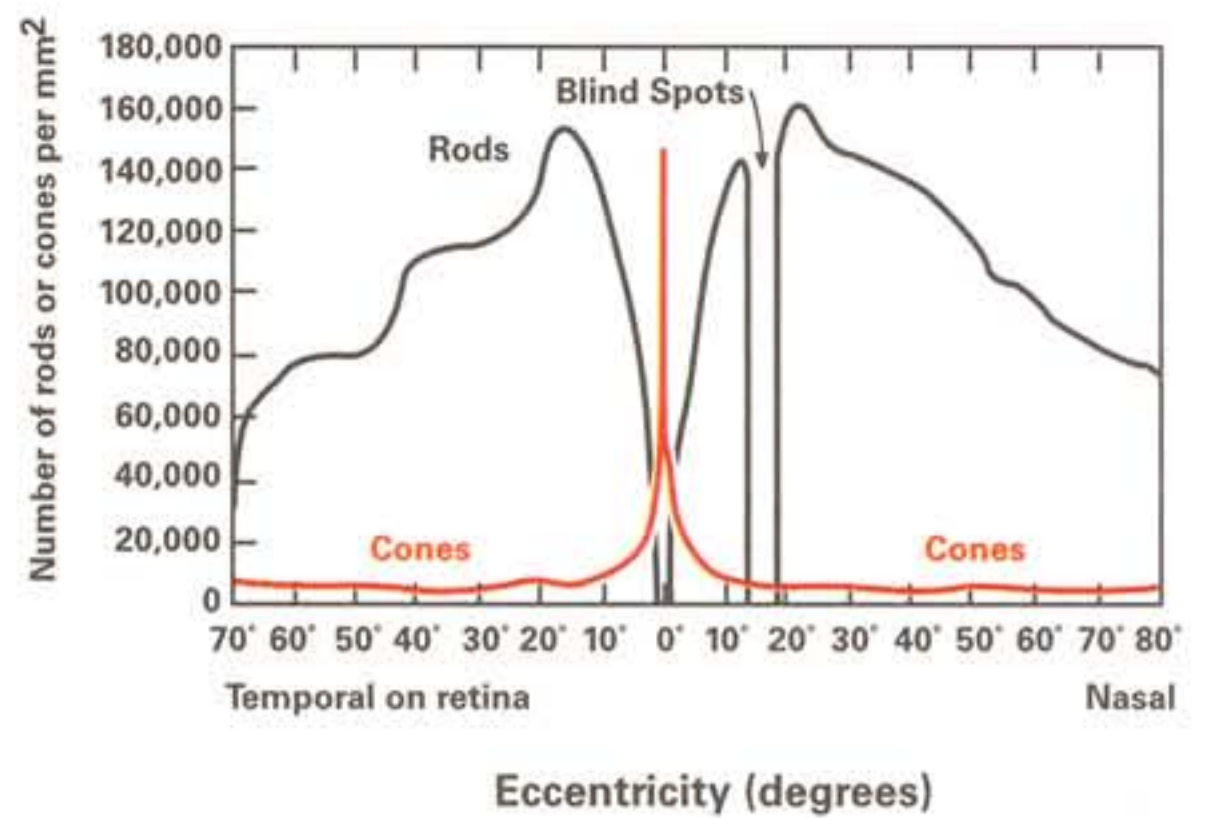


FIGURE 1.28 Distribution of photoreceptors in the human eye. By permission of Resnikoff (1987, Figure 5.3.3).

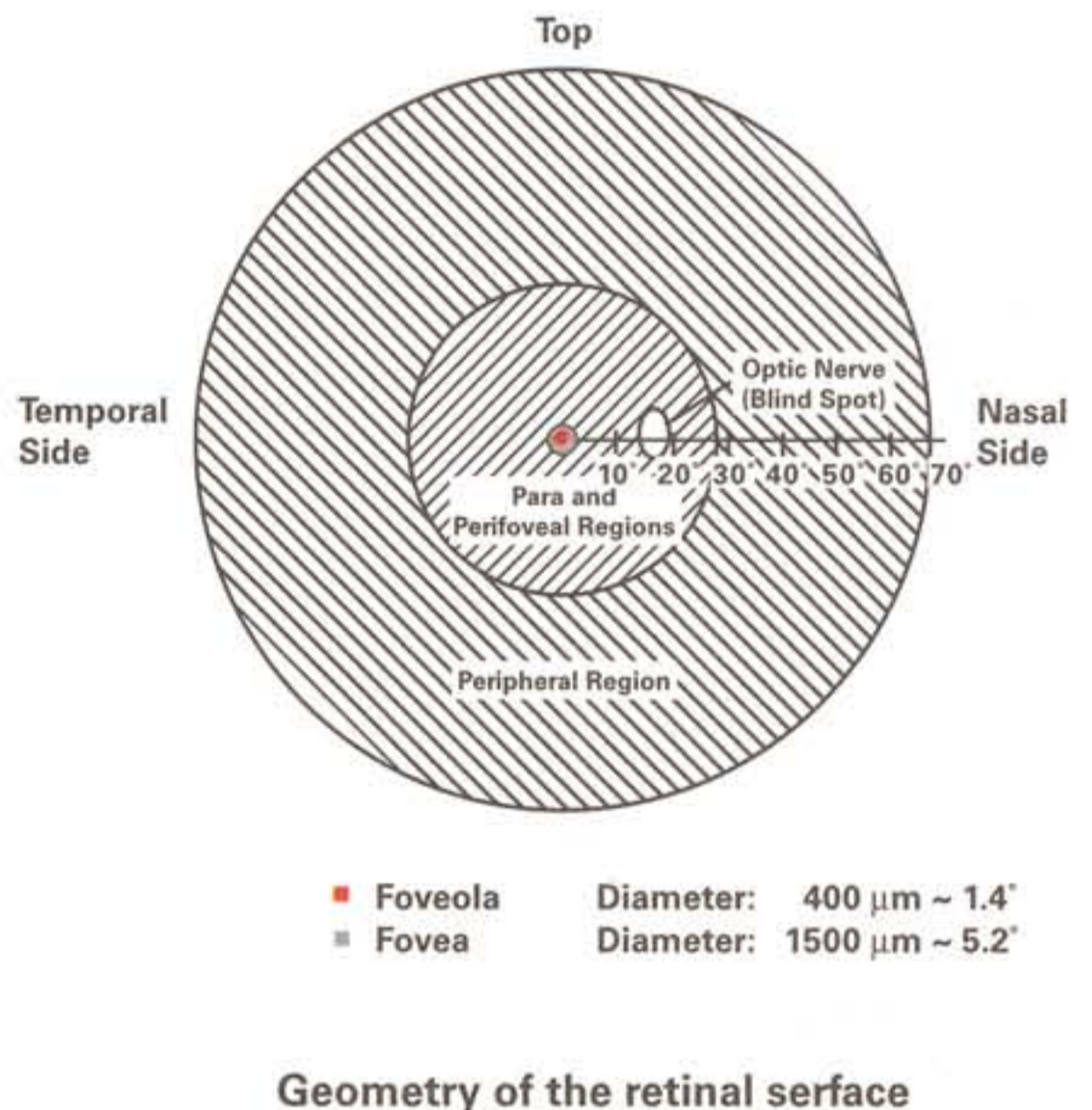


FIGURE 1.29 Logical map of photoreceptors in the eye. By permission of Resnikoff (1987, Figure 5.3.4).

movement or other changes in the visual environment and in visually maintaining a rough representation of the location of shapes previously examined. Just how little detail is available peripherally can be seen in Figure 1.30, a photograph of a scene processed to simulate the information available in the various parts of the visual field.

The second level of the visual system is approximately the *foveola* (the inner part of the fovea), the  $400 \mu\text{m}^2$  (about  $1.4^\circ$ ) in the center of the visual field. The entire retinal field is the equivalent of  $7950 \approx 8000$  foveolae. This high-resolution field is moved to points of interest about 1 to 5 times/sec at rates of up to  $500^\circ/\text{sec}$ , during which vision is suppressed.

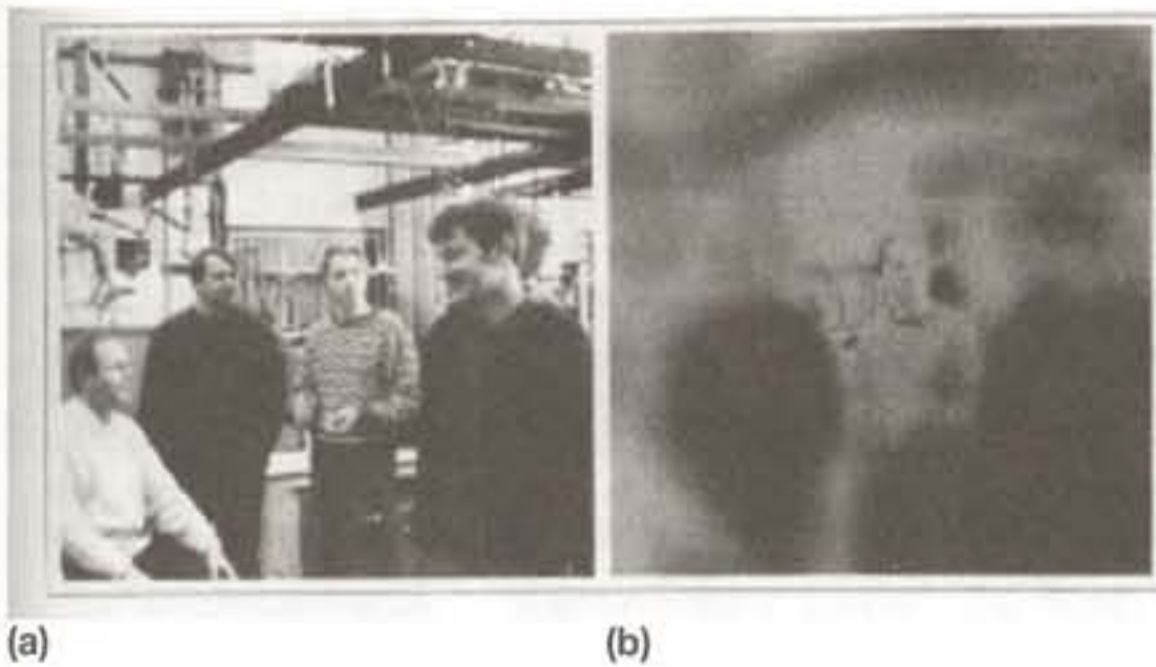


FIGURE 1.30

The visual field at any instant in time. The photograph in (a) has been processed to simulate in (b) the level of detail available at different places in the visual field. While little detail can be seen in the periphery, the general shapes and positions are preserved (Tovée, 1996, Figure 10.1).

(The eye also has tiny movements—as many as 70 times/sec—and slow pursuit movements that keep images steady.) In fact, the eye movement mechanism is part of a more complex attention mechanism including head movements and a variable-size attention window working on the visual buffer (Kosslyn, 1994). Automatic, stimulus-based attention shifting causes this mechanism to shift toward either movement or areas where preattentive features have identified strong patterns of color, intensity, or size contrast. Stereoscopic processing on the differences between the images of the two eyes gives depth information, as does *head parallax*, the moving of the head to disambiguate images. Information on configurations of interest are sent to two separate systems, one that encodes spatial properties such as location, size, and orientation, and another that encodes object properties such as shape, color, and texture (Kosslyn, 1994).

The third level of the hierarchical visual system is the set of receptors themselves within the foveola. In the foveola, the density of cones is something like 27 times greater than in the periphery. In fact, since the number of cones per neuron is around 8:1 in the periphery versus 1:1 in the foveola, the information density may be as much as 200 times greater (Resnikoff, 1987).

Thus, the system maintains a constant, computationally parallel surveillance over the entire visual field. At the same time, it is constantly moving the position of the foveola, sampling from the visual field to build up a percept or to attend to areas of high information content, such as moving objects. Visual perception is an active process in which head, eye, and attention are all employed to amplify information per unit time from the visual world.

The visual system does not work like a photograph developing in a camera but like a flying-spot scanner. It trades off time resolution to reduce the bandwidth by something like a factor of 8000 foveolae equivalents  $\times$  200 times greater information density =  $1.6 \times 10^9$  (or put differently, it increases the resolution for a given available bandwidth). The visual system knits together a remarkable illusion of continuity from

the succession of saccades, extracting interpretations from high-information features like sharp corners and gestalt continuity, and making invisible the missing array of receptors where the optic nerve is attached (the “blind spot”).

To get a sense of how different a percept is than a photograph, imagine a person driving a car down the freeway. The driver looks ahead, into the rearview mirror, and occasionally to the side, aware of the traffic ahead, that there is a car too close behind, that another is passing on the right. At any particular moment, the driver *perceives* more than he or she instantaneously *sees*, because the percept of the traffic situation is built up from discrete visual samples of the environment. In fact, the driver will tend to sample the different visual sources roughly proportionally to the amount of information contained in them (if there is not an information overload). A car changing lanes will get more attention than one whose relative position is constant.

Visual information can be processed in two different ways, sometimes called controlled and automatic processing. *Controlled processing*, like reading, uses mainly the fovea. The processing is detailed, serial, low capacity, slow, able to be inhibited, conscious. *Automatic processing* in contrast is superficial, parallel, can be processed nonfoveally, has high capacity, is fast, cannot be inhibited, is independent of load, unconscious, and characterized by targets “popping out” during search. Actually, the contrast is not quite so crisp as this comparison suggests (see Shiffrin, 1988), but the general distinction is still important and practical. While visualizations can be designed so that detail, such as textual description, is accessible by controlled processing, coding techniques to aid search and pattern detection should use features that can be automatically processed. Color and size are typical features used to code data visually in a form capable of automatic processing, but the literature suggests more exotic features as well (these are discussed later on in Table 1.22). Many of these coding methods have not yet been tried, but because they are known to be automatically processable, they are candidates for constructing new visualization techniques.

There can be interaction among the visual codings of information. Indeed, part of the point of coding information visually is to produce patterns that the eye detects from ensembles of components. If these interactions are unintended, however, the user will be misled. The gestalt principles shown in Table 1.20 collect some well-known interactions. For example, objects near each other will tend to be seen as a cluster. Causing related objects to cluster tightly enough for this visual effect to occur may be a reason for choosing a particular layout algorithm. Eick and Wills (1993 ●), for example, argue that the “spring model” for object layout on a display is not as good as their own model, because it makes groups harder to spot.

The fact that human perception divides into focus and periphery can be exploited, not just in coding objects but also in setting up visual frames that serve as a substrate for the encoding of objects and patterns. As objects are examined, their locations become visually indexed so that search



TABLE 1.20

Gestalt principles of organization. After Tové (1996, Table 8.2). Used with permission.

RULE	BOUNDARIES
Pragnanz	Every stimulus pattern is seen in such a way that the resulting structure is as simple as possible.
Proximity	The tendency of objects near one another to be grouped together into a perceptual unit.
Similarity	If several stimuli are presented together, there is a tendency to see the form in such a way that the similar items are grouped together.
Closure	The tendency to unite contours that are very close to each other.
Good continuation	Neighboring elements are grouped together when they are potentially connected by straight or smoothly curving lines.
Common fate	Elements that are moving in the same direction seem to be grouped together.
Familiarity	Elements are more likely to form groups if the groups appear familiar or meaningful.

time to relocate them is reduced. The dimensions of space or patterns on the space itself, such as lines joining nodes, may be assigned meanings. As a result, objects may form a spatial external working memory. Enlarging working memory can lead to dramatic improvements of cognitive functions (see, e.g., Figure 1.1). Visualizations can also be used to store large numbers of detailed facts for rapid access (e.g., the periodic table or a ship chart).

### Spatial Substrate

Not only are there characteristic limits to the perceptual system, there are also representational limits to graphics as a medium. The number of basic mappings of Data Tables to Visual Structures is actually smaller than might be supposed, because there are a limited number of components from which Visual Structures are composed. Visual Structures are made from *spatial substrate*, *marks*, and the marks' *graphical properties* (Mackinlay, 1986a). This limited set was identified by Bertin (1977/1981), expanded by Mackinlay (Card and Mackinlay, 1997; MacEachren, 1995; Mackinlay, 1986b •), and expanded further here. Other properties, as we shall argue, are possible, but most visualizations will probably continue to be made from this basic set.

The most fundamental aspect of a Visual Structure is its use of space. Space is perceptually dominant (see MacEachren, 1995). Spatial position is such a good visual coding of data that the first decision of visualization design is which variables get spatial encoding at the expense of others. One reason for the effectiveness of Tufte's *Challenger* diagram is that he maps the most important variables onto spatial position in X and Y, the most potent representation properties of the Visual Structure. Like other visual features, spatial position can be used to encode the variables of Data Tables. But because of its dominance, we treat it separately from these other features as a substrate into which other parts of a Visual Structure are poured.

Empty space itself, as a container, can be treated as if it has metric structure. We describe this structure in terms of axes and their properties. There are four elementary types of axes:

U = *Unstructured Axis* (no axis) (Engelhardt et al., 1996),

N = *Nominal Axis* (a region is divided into subregions),

O = *Ordinal Axis* (the ordering of these subregions is meaningful), and

Q = *Quantitative Axis* (a region has a metric).

Further subdivision of the quantitative axis is possible, namely, whether the quantitative axis has interval or ratio properties. There are also important specializations to physical coordinates (a quantitative axis with physical units) or geographical coordinates (the specialized physical coordinates of latitude and longitude). But this simple division suffices for our present purposes. Axes can be linear or radial.

Axes are an important building block for developing Visual Structures. The FilmFinder (Ahlberg and Shneiderman, 1994b) in Figure 1.31 augments a scatterplot with a collection of user interface sliders and radio buttons. These allow rapid query specification through direct manipulation, which is coupled with instantaneous feedback. Based on the Data Table for the FilmFinder in Table 1.6, we represent the scatterplot as composed of two orthogonal quantitative axes:

Year  $\rightarrow Q_x$ ,  
Popularity  $\rightarrow Q_y$ .

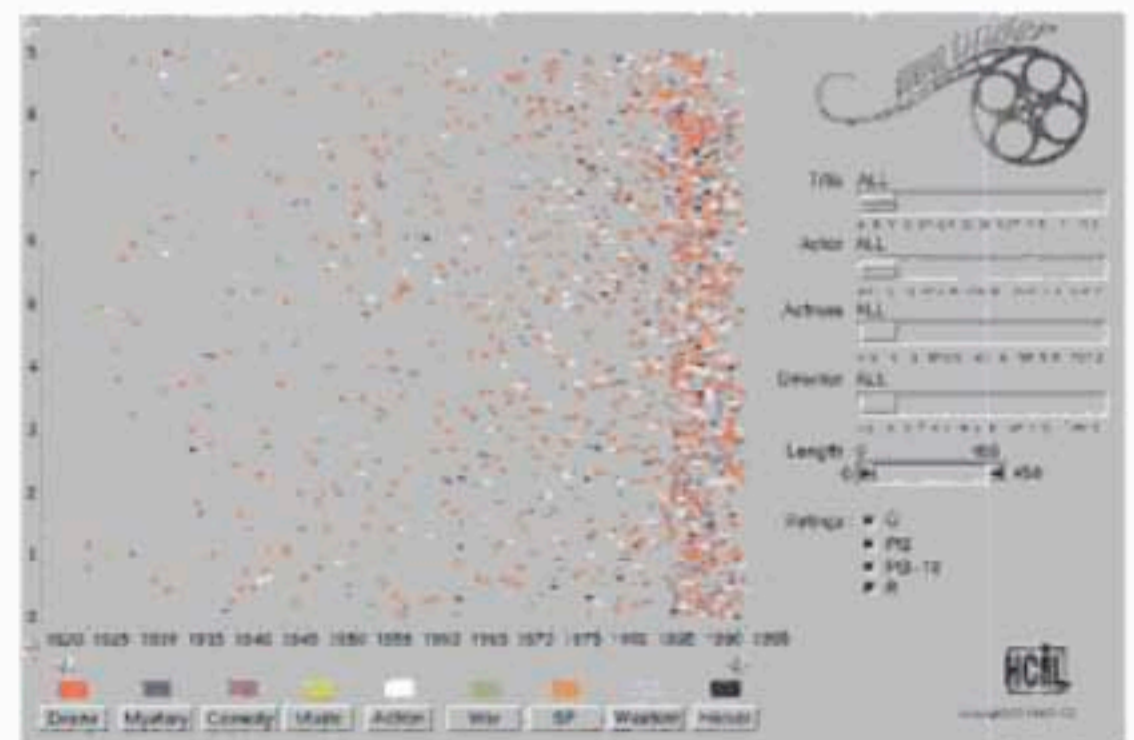


FIGURE 1.31

The FilmFinder. Courtesy of the University of Maryland. See Ahlberg and Shneiderman (1994b).

The notation states that the *Year* variable is mapped to a quantitative X-axis and the *Popularity* variable is mapped to a quantitative Y-axis. Information is encoded by mapping the cases, which are represented by the *FilmID* variable, to points:

$$FilmID \rightarrow P$$

Positioning these points on the axes:

$$FilmID(Year, Popularity) \rightarrow P(Q_x, Q_y)$$

encodes the year and popularity of the films.

Other axes are used for the FilmFinder query widgets. For example, an ordinal axis is used in the radio buttons for film ratings,

$$Ratings \rightarrow O_y$$

A nominal axis is used in the radio buttons for film type,

$$FilmType \rightarrow N_x$$

Since spatial position is such a good encoding, several techniques have been developed to increase the amount of information that can be encoded with it:

- Composition
- Alignment

- Folding
- Recursion
- Overloading

*Composition* (Mackinlay, 1986b •) is the orthogonal placement of axes, creating a 2D metric space. The FilmFinder scatterplot in Figure 1.31 creates such a space where a person directly perceives relationships between film popularity and their year of production. This technique is powerful for up to two variables and still potent up to three dimensions. Even at three dimensions, if the content of the resulting cube is dense, we have the problem of seeing inside.

*Alignment* (Mackinlay, 1986b •) is the repetition of an axis at a different position in the space. For example, the bond market visualization in Figure 1.13 shows the alignment of two Visual Structures on a common X-axis, representing time. The Visual Structure on the floor representing individual bond performance is aligned with the yield curve on the back wall.

*Folding* is the continuation of an axis in an orthogonal dimension. Figure 1.32 is a visualization of a large computer program. Each software module is represented as an axis consisting of line marks to represent the text lines of the program. These axes (oriented in the Y-direction) are folded when they are too long to fit in the window by using space

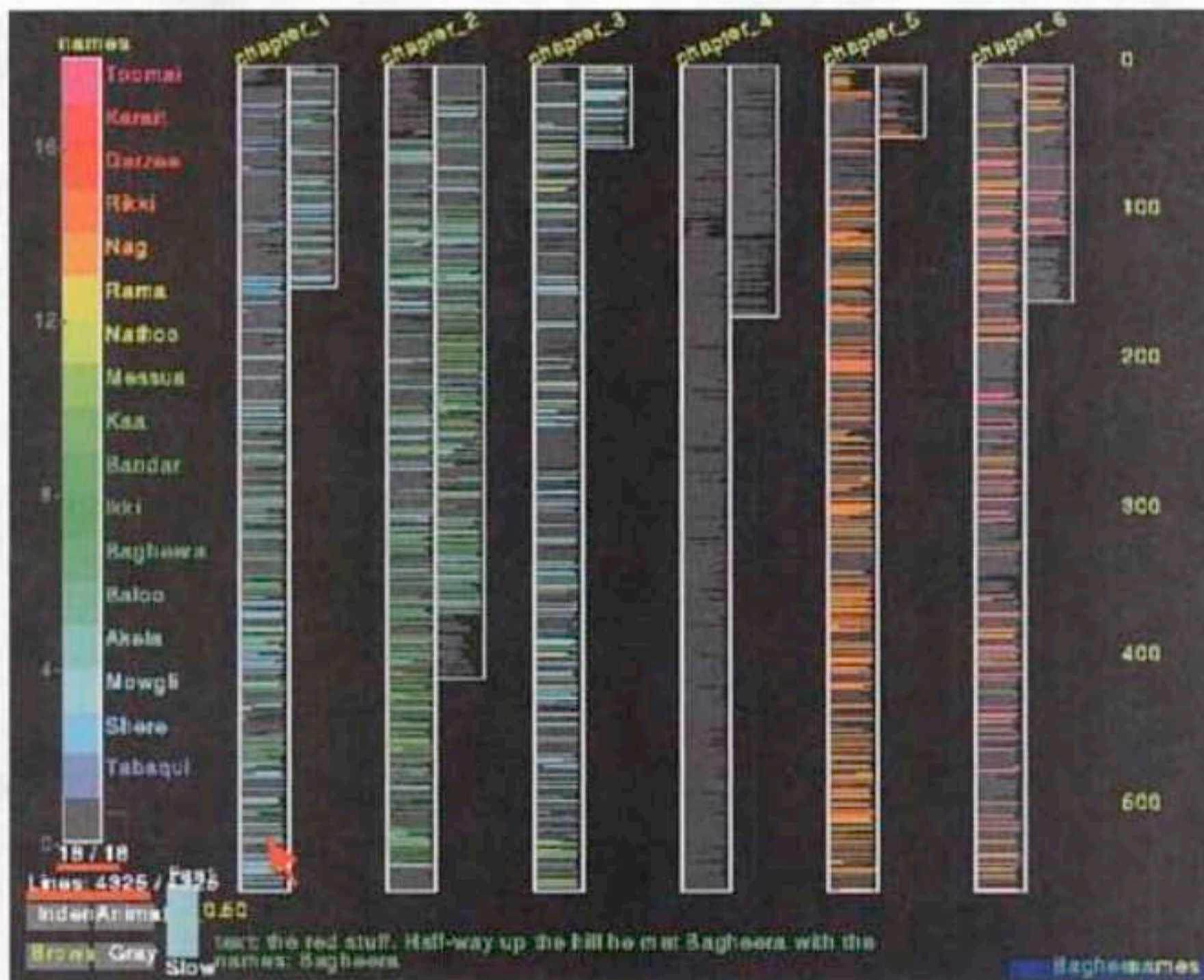


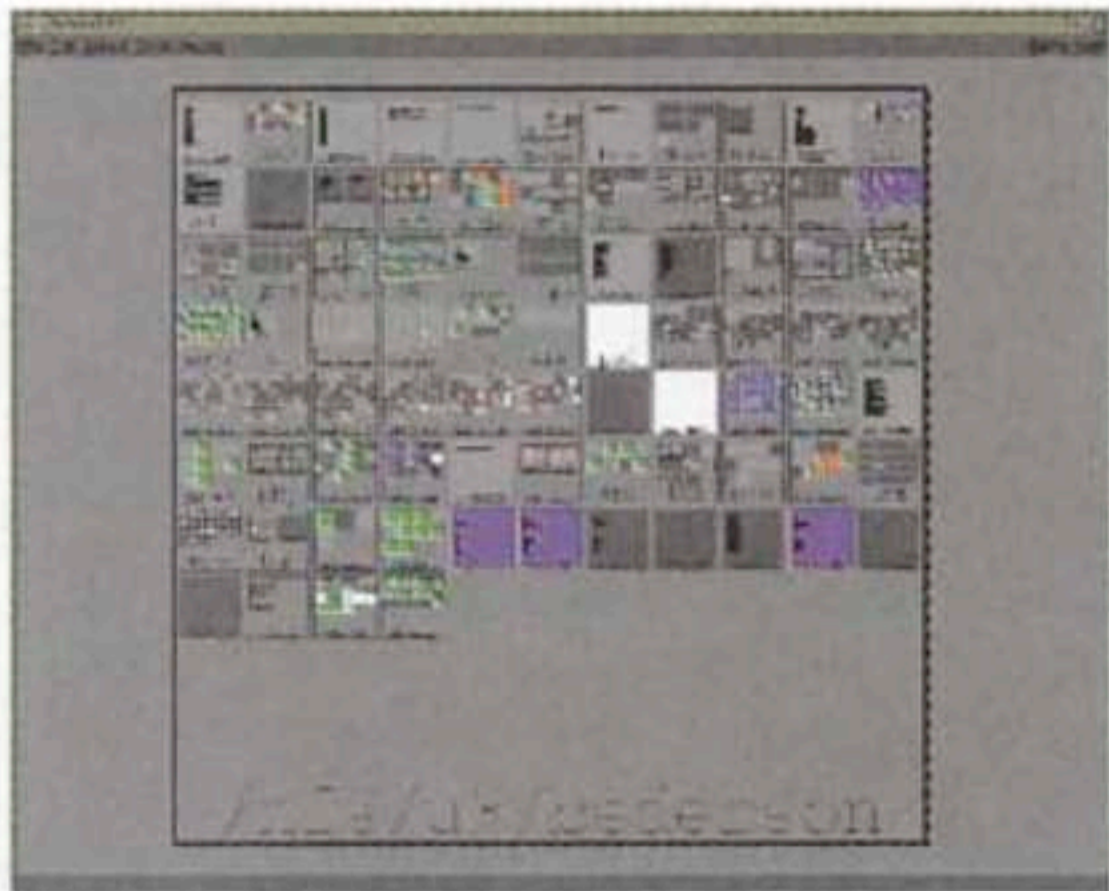
FIGURE 1.32

SeeSoft uses a folded axis when a software module is too large to fit in the height of the window. Courtesy of Lucent Technologies. See Eick, Steffen, and Sumner (1992 •). Used with permission of Lucent Bell Laboratories.

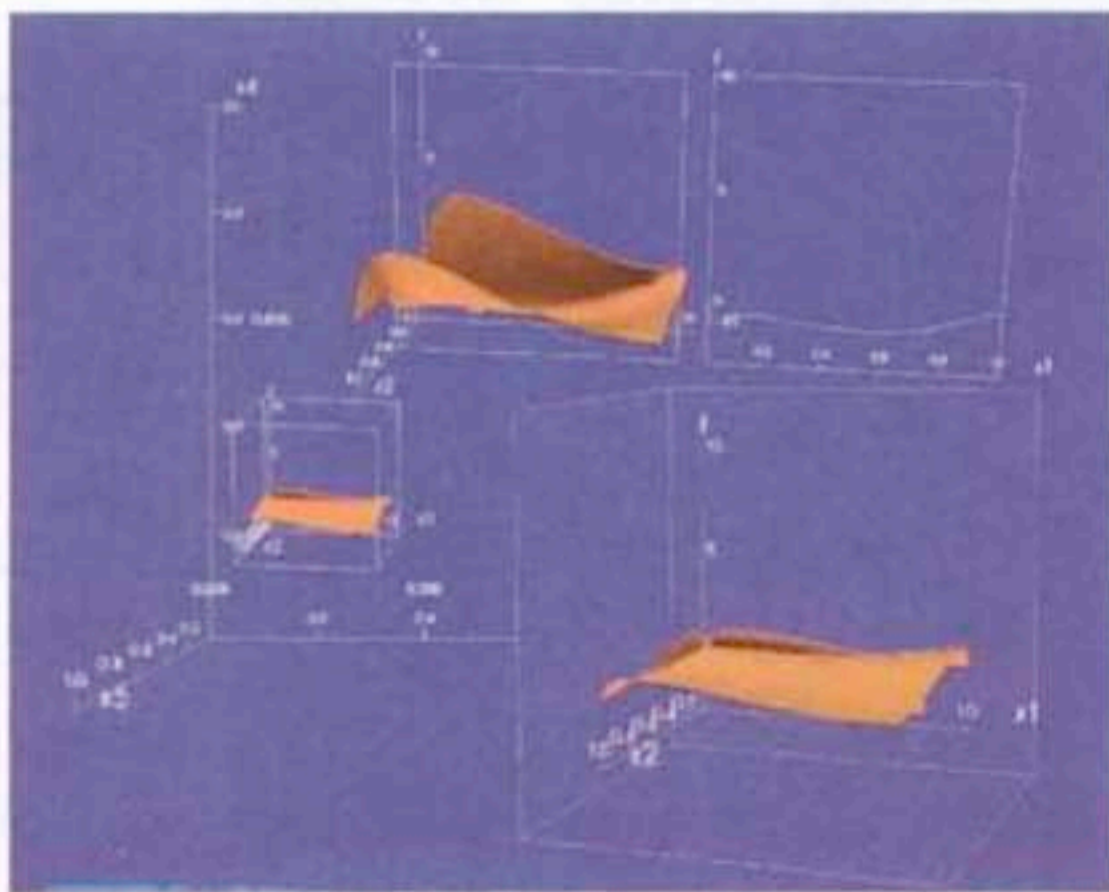
offset in the X-direction from the already used space. This visualization is also an example of axis alignment because of the alignment of the ordinal position of the text lines.

*Recursion* is the repeated subdivision of space. Figure 1.33 is a screen shot from Pad++ (Bederson and Hollan, 1994 ●) that provides interactive zoom into a recursive space of directories and files. A folded axis creates the top-level partitioning of the space into a set of rectangles that represent directories. Inside each of these regions are additional axes that recursively partition the space.

*Overloading* is the reuse of the same space for the same Data Table. In the worlds within worlds technique (Feiner and Beshers, 1990b ●), shown in Figure 1.34, the meaning



**FIGURE 1.33**  
Pad++ provides interactive zoom into a recursive space of directories and files. Courtesy of Jim Hollan. See Bederson and Hollan (1994 ●).



**FIGURE 1.34**  
Worlds within worlds (Feiner and Beshers, 1993, Figure 2) overloads space to visualize multivariable data tables.

of one coordinate system is determined by its placement inside another. The technique plays heavily on the fact that the data occupies only a portion of the committed space, allowing that space to be recommitted to a second use. Because this overloading is dynamically controlled by the user in this application, the user may be willing to accept some occlusion.

**Marks**

*Marks* are the visible things that occur in space. There are four elementary types of marks (Figure 1.35):

- P = *Points* (0D or zero dimensional),
- L = *Lines* (1D),
- A = *Areas* (2D), and
- V = *Volumes* (3D).

Area marks include surfaces in three dimensions as well as 2D-bounded regions.

Unlike their mathematical counterpart, point and line marks actually take up space (otherwise they would be invisible) and may have properties like shape. They take up space to signify something that does not.

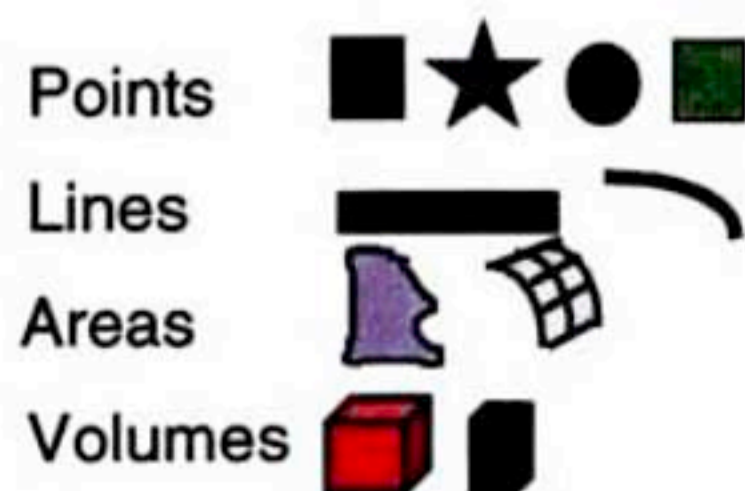
**Connection and Enclosure**

Point marks and line marks can be used to signify another sort of topological structure: *Graphs* and *Trees*. These allow relations among objects (e.g., Table 1.10) to be shown without the geometrical constraints implicit in mapping variables onto spatial axes:

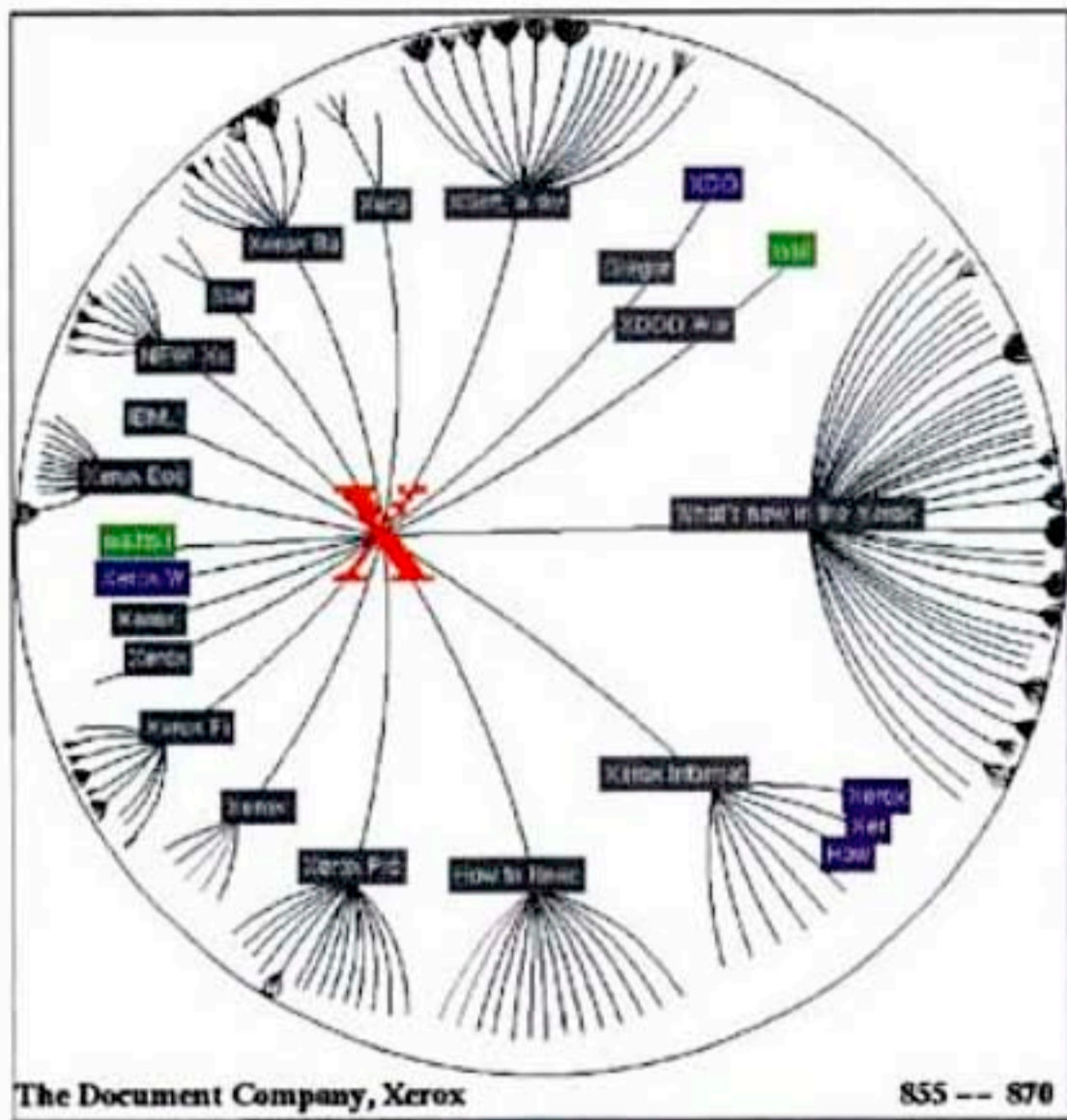
*Links* → *Connection*.

Figure 1.36 is a screen shot of the hyperbolic tree (Lamping and Rao, 1996 ●), a visualization that uses a hyperbolic projection to show more detail in the vicinity of some focal point. The position of the nodes is used to make the objects more visually salient rather than encoding information directly.

Trees and graphs also use position to create gestalt properties such as proximity or closure (see Table 1.20). Because these are easily picked up as perceptual features, they can encode additional information such as clustering or partial trends. Trees typically start with a root node and continue with levels that represent the generations of children nodes.



**FIGURE 1.35**  
Types of marks.



**FIGURE 1.36**  
Hyperbolic tree. See Lamping and Rao (1996). Courtesy of Xerox Corporation.

These levels form an implicit ordinal axis that encodes the distance of a node to the root even when the Raw Data does

not include this information explicitly, as in the radial axis in the hyperbolic tree. Constellations of data relations can trigger these as emergent visual properties, signaling the existence of the underlying data relation. However, as we have noted, care must be taken not to inadvertently express incorrect information (Mackinlay, 1986b).

Enclosure can also be used to encode hierarchies:

*Links* → Enclosure.

Figure 1.37 is a treemap (Johnson and Shneiderman, 1991), mapping a library system into nested rectangles. The size of the rectangles is determined by the number of books. The hierarchy determines the nesting. Color indicates frequency of use (redder is more frequent).

**Retinal Properties**

Other graphical properties were called *retinal properties* by Bertin (1967/1983), because the retina of the eye is sensitive to them independent of position. For example, the Film-Finder in Figure 1.31 uses color to encode information in the scatterplot:

$FilmID(FilmType) \rightarrow P(Color)$

This notation says that the *FilmType* attribute for any *FilmID* case is visually mapped onto the color of a point.

Table 1.21 shows Bertin's six "retinal variables" separated into spatial properties and object properties according to



**FIGURE 1.37**  
Treemap of Dewey decimal classification. Courtesy of the University of Maryland.

TABLE 1.21

Retinal properties.

	Spatial	Object
Extent	(Position) — — —	Gray Scale ■ ■ ■ □
	Size ● ● ● ●	
Differential	Orientation — /   \	Color ■ ■ ■ ■
		Texture ■ ■ ■ ■
		Shape ■ ★ ● ◆

which area of the brain they are believed to be processed (Kosslyn, 1994). They are cross-separated according to whether the property is good for expressing the extent of a scale (has a natural zero point) or whether its principal use is for differentiating marks (Bertin, 1977/1981). Spatial position, discussed earlier as basic visual substrate, is shown in the position it would occupy in this classification.

Other graphical properties have also been proposed for encoding information. For example, MacEachren (1995) proposes *crispness* (the inverse of the amount of distance used to blend two areas or a line into an area), *resolution* (grain with raster or vector data will be displayed), *transparency*, and *arrangement* (e.g., different ways of configuring dots). He further proposes dividing color into *value* (essentially the *gray level* of Table 1.21), *hue*, and *saturation*. The usefulness of these requires testing. On the other hand, graphical properties from the perception literature that can support automatic visual processing (or at least preattentive processing) are other obvious candidates for coding variables. Several of these are collected in Table 1.22 from Healy, Booth, and Enns (1995). For example, lighting direction might be usable as a visual coding dimension in a Visual Structure, although to our knowledge this has not yet been attempted. We will use the retinal properties in Table 1.21 because they are a good basic set for our purposes, but it should be remembered that there are other possibilities.

Some retinal properties are more effective than others for encoding information. Position, for example, is by far the most effective all-around representation. Many properties are more effective for some types of data than for others. Grayscale, for example, is effective when used comparatively for ordinal variables, but is not very effective for encoding

TABLE 1.22

Visual features that can be automatically processed (Healy, Booth, and Enns, 1995).

Number	Terminators	Direction of motion
Line orientation	Intersection	Binocular luster
Length	Closure	Stereoscopic depth
Width	Color	3D depth cues
Size	Intensity	Lighting direction
Curvature	Flicker	

TABLE 1.23

Relative effectiveness of different retinal properties. Data based on MacEachren (1995, Figure 6.30). Q = Quantitative data, O = Ordinal data, N = Nominal data. Filled circle indicates the property is good for that type of data. Half-filled circle indicates the property is marginally effective, and open circle indicates it is poor.

	Spatial	Q	O	N	Object	Q	O	N
Extent	(Position)	●	●	●	Grayscale	◐	●	○
	Size	●	●	●				
Differential		◐	◐	●	Color	◐	◐	●
	Orientation				Texture	◐	◐	●
					Shape	○	○	●

absolute quantitative variables. Table 1.23 gives the relative effectiveness of different retinal properties.

### Temporal Encoding

Visual Structures can also encode information temporally: Human perception is very sensitive to *changes* in mark position and their retinal properties. We need to distinguish between temporal Data Tables that need to be visualized, as in

$$Q_t \rightarrow \text{some visual representation}$$

and animation, that is, time used as part of a Visual Structure:

$$\text{some variable} \rightarrow \text{Time.}$$

Time as animation could encode any type of data (whether it would be an effective encoding is another matter).

Time as animation, of course, can be used to visualize time data:

$$Q_t \rightarrow \text{Time.}$$

This is natural but not always the most effective encoding. Mapping time data into space allows comparisons between two points in time. For example, if we map time and a function of time into space (e.g., time and accumulated rainfall),

$$Q_t \rightarrow Q_x \text{ [make time be the X-axis]}$$

$$f(Q_t) \rightarrow Q_y \text{ [make accumulated rainfall be the Y-axis]}$$

then we can directly experience rates as visual linear slope, and we can experience changes in rates as curves. Tufte (1994) shows a more sophisticated variant in which miniature visualizations are arranged along an axis of time. This display then becomes a control for controlling an animated sequence. Another use of time as animation is similar to the unstructured axes of space. Animation can be used to enhance the ability of the user to keep track of changes of view or visualization. If the user clicks on some structure causing it to enlarge and other structures to become smaller, animation can effectively convey the change and the identity of objects across the change, whereas simply viewing the two

end states is confusing. Another use is to enhance a visual effect. Rotating a complicated object, for example, will induce 3D effects (and hence allow better reading of some visual mappings).

## VIEW TRANSFORMATIONS

*View transformations* interactively modify and augment Visual Structures to turn static presentations into visualizations by establishing graphical parameters to create Views of Visual Structures. Visualizations exist in space-time. View transformations exploit time to extract more information from the visualization than would be possible statically. There are three common view transformations:

1. Location probes
2. Viewpoint controls
3. Distortions

### Location Probes

*Location probes* are view transformations that use location in a Visual Structure to reveal additional Data Table information. Figure 1.38 shows the FilmFinder after the user probes a point in the scatterplot. The resulting details-on-demand pop-up window gives details about the film mapped to the point. Brushing is a form of probe where the cursor passing over one location creates visual effects at others' marks (McDonald, 1990).

Probes can also augment the Visual Structure. Scientific visualizations use slicing plane probes to access the interior of 3D solid objects (DeFanti, Brown, and McCormick, 1989). Streamlines are a probe that renders vector fields visible. *Magic lenses* (Fishkin and Stone, 1995) are probes that give an alternate view of a region in the Visual Structure. Objects in the region reveal additional properties of the Data Table.

### Viewpoint Controls

*Viewpoint controls* are view transformations that use affine transformations to zoom, pan, and clip the viewpoint. These transformations are common, because they magnify Visual Structure or change the point of view, which makes the details more visible. For example, Figure 1.38 shows the FilmFinder zoomed into a small part of the scatterplot.

The problem with zooming is that the surrounding area (the context) disappears as the details are zoomed. One strategy, explored by the Pad (Perlin and Fox, 1993) and Pad++ systems (see Figure 1.33), is to make the zoom rapid and easy to invoke (they assign it to mouse buttons) (Bederson and Hollan, 1994 •). However, this requires the user to remember information not visible.

Another viewpoint control technique is called *overview + detail* (Shneiderman, 1996). Two windows are used together: an overview of the Visual Structure and a detail window that provides a magnified focus for one area. The overview provides a context for the detail view and acts as a control wid-

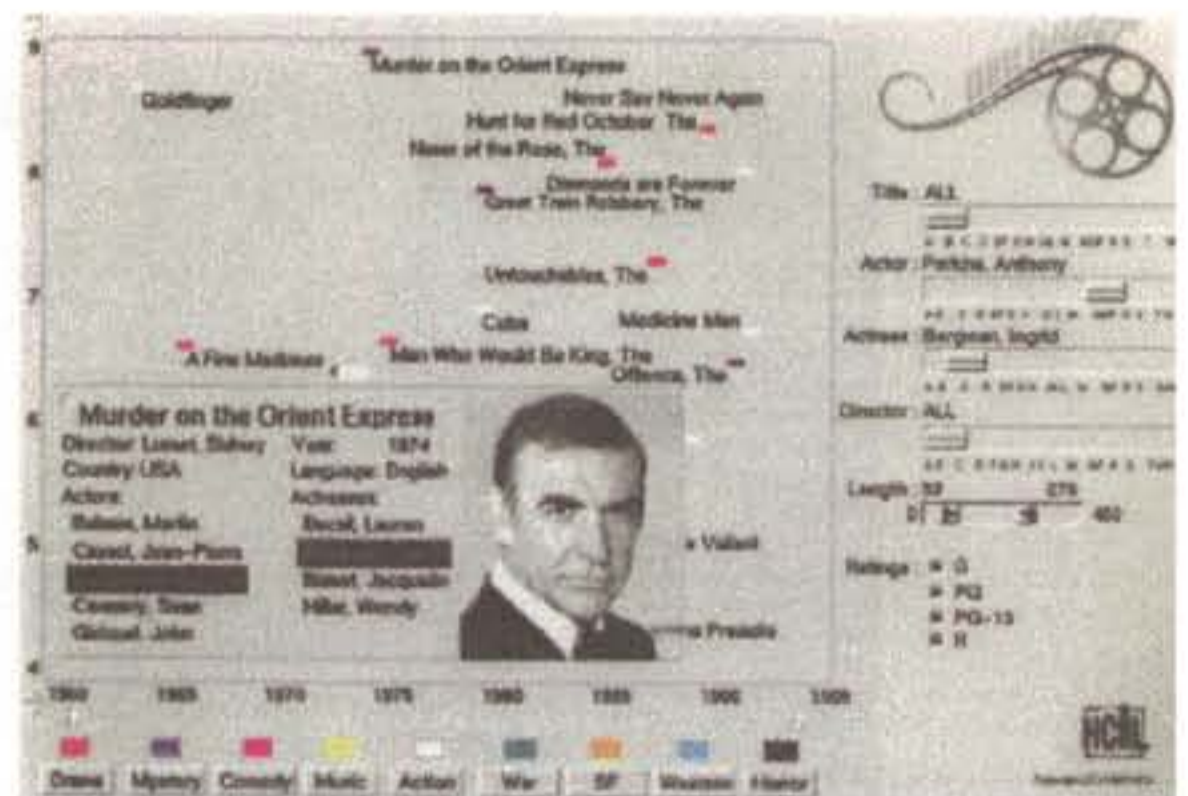


FIGURE 1.38

FilmFinder showing the details of a probed film. Courtesy of the University of Maryland. See Ahlberg and Shneiderman (1994a).

get to change the detail view. Figure 1.39 shows a visualization of an algorithm using an Information Mural (Jerding and Stasko, 1995a). The lower window gives an overview of the entire set of messages. The upper window shows the detail in the area indicated by the rectangle in the lower window. The message types are associated with a color resulting in characteristic color patterns in the overview window. Zoom factors of between 5 and 30 seem to work best, with larger zoom factors requiring an intermediate view (Shneiderman, 1998; Plaisant, Doan, and Shneiderman, 1995).

The overview + detail technique has both strengths and weaknesses. One strength is that it is simple to implement and understand. Another strength is that it can provide rapid access to the details of a visualization that is too large to fit on a computer display. Its primary weakness is that comparison may require the movement of the detail window, including disrupting shifts of attention to the overview window. Overview analysis may require Visual Structures that do not fit in the overview window, which is typically much smaller than the detail window.

### Distortion

*Distortion* is a visual transformation that modifies a Visual Structure to create focus + context views. Overview and detail are combined into a single Visual Structure. The *hyperbolic tree* (Figure 1.36) distorts a large tree layout (actually it distorts the space on which the tree is laid out) with a hyperbolic transformation that maps a plane to a circle, shrinking the nodes of the tree far from the root. The *perspective wall*, shown in Figure 1.40, shows when files in a computer system were modified. Clicking on the file symbols in the bent part of the wall slides the wall so as to bring them into the central focus area.

Distortion is effective when the user can perceive the larger undistorted Visual Structure through the distortion. For example, the *bifocal lens* (Spence and Apperley, 1982 •) supports the perception of linear sequence, although objects

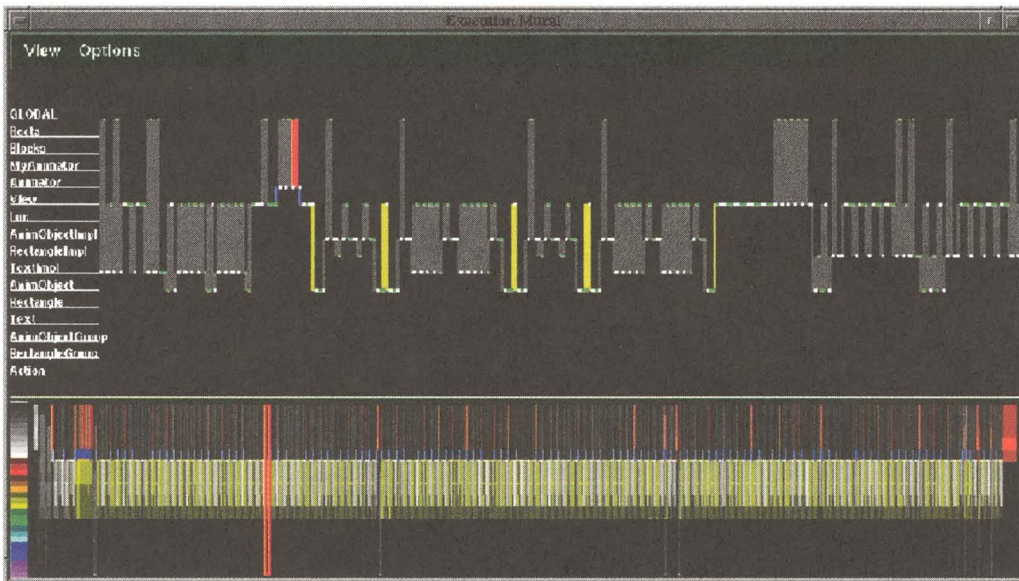


FIGURE 1.39

Information Mural (Jerding and Stasko, 1995a, Figure 2) used overview + detail to view a long sequence of messages in a program performing a bubble sort.

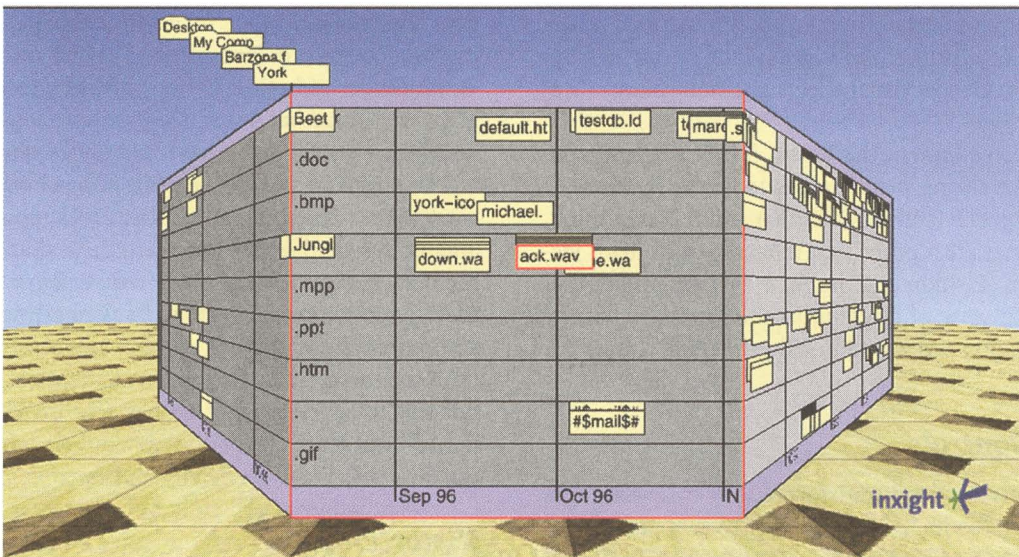


FIGURE 1.40

Perspective wall. Courtesy of Inxight Software and Xerox Corporation. See Mackinlay, Robertson, and Card (1991).

outside the focal area have distorted aspect ratios. Distortions can be roughly classified by what the human perceives as invariant. The perspective wall (Mackinlay, Robertson, and Card, 1991) is similar to the bifocal lens, but the human perceives the linear sequence as folded, which means it is a distortion that leaves even the metric information invariant (Mackinlay, 1986b •). The bifocal lens is an example of a 1D distortion that leaves ordering invariant. The

*table lens* (Rao and Card, 1994 •) is an example of a 2D distortion that leaves ordering invariant. Three-dimensional distortions are also possible (Carpendale, Cowperthwaite, and Fracchia, 1997 •). The next most general type of distortion leaves topological relationships invariant, e.g., the hyperbolic tree (Lamping and Rao, 1996 •). Distortion is not effective when the features or patterns of use to the user are distorted in a way harmful to the task.

## INTERACTION AND TRANSFORMATION CONTROLS

The final part of our reference model (Figure 1.23) is human interaction, completing the loop between visual forms and control of visualization parameters in the service of some task. The most obvious form of interaction is direct manipulation. For example, the nodes in a hyperbolic tree (Figure 1.36) can be dragged with the mouse to the center of the display.

Interaction includes techniques for controlling mappings in Figure 1.23:

*Raw Data → Data Table.* The FilmFinder (Figure 1.31) is an example of the interactive control of data mappings. The sliders filter cases from the complete Data Table of films, selecting those that appear in the Visual Structure scatterplot. The resulting query is a conjunct of ranges specified using the user interface widgets shown in Figure 1.31. The resulting tight coupling between query and result is more effective than entering query commands.

*Data Table → Visual Structure.* Interactive control of the mapping from Data Table to Visual Structure can be provided in a separate user interface or integrated with the Visual Structure. Many scientific visualization systems use a separate dataflow window for their controls. Data Tables and Visual Structure are represented in this window as rectangles that have input and output spots. The user controls the mapping by connecting inputs to outputs. In contrast, integrated techniques allow the user to click on parts of the Visual Structure to change the mapping. In the FilmFinder, the user might click on the Y-axis to change *Popularity to Rating*.

*Visual Structure → View.* Interactive control of the view can also be a separate or integrated interface. Probes and viewpoint manipulations are typically integrated. Distortion techniques often have a more global impact that may require an external user interface, but they can be integrated. For example, the table lens provides small handles on the focal region for making changes.

## CONCLUSION

The reference model of information visualization developed in this chapter approximates the basic steps for visualizing information: The first step is to translate Raw Data to a Data Table, which can then be mapped fairly directly to a Visual Structure. View transformations are used to increase the amount of information that can be visualized. Human interaction with these Visual Structures and the parameters of the mappings create an information workspace for visual sense making.

In real life, visual sense making usually combines these steps into complex loops. Human interaction with the information workspace reveals properties of the information that lead to new choices. Designing means for carrying out these mappings leads to a number of techniques. Table 1.24 lists some of these in summary. The rest of this book collects examples in detail.

In the papers that follow, we use the reference model to follow the literature in this newly emerging area. Chapter 2 surveys mappings of abstract data into spatial form. Chapter 3 considers methods for interacting with these mappings.

TABLE 1.24

The components of the reference visualization model shown in Figure 1.23. Specific techniques are also included in the table. The specific techniques for Data Tables, discussed in the text, are a list of common data types that have well-known Data Tables. Tasks are operations that a user may want to do with the visualization.

DATA TABLES	VISUAL STRUCTURES	VIEWS	HUMAN INTERACTION	TASKS	LEVEL
Cases Variables Values Metadata	Spatial Substrate Marks Graphical properties	Location Probes Viewpoint Controls Distortion	Data Tables Visual Structures Views	Forage for Data Problem Solving Search for Schema Instantiate Schema Author, Decide, or Act	Infosphere Workspace Visual Knowledge Tools Visual Objects

### Specific Techniques

Spatial (Scientific) Geographic Documents Time Database Hierarchies Networks World Wide Web	Position: NOQ Marks: PLAV Properties: Connection, Enclosure, Retinal, Time Axes: Composition Alignment Folding Recursion Overloading	Brushing Zooming Overview + Detail Focus + Context	Dynamic Queries Direct Manipulation Magic Lens	Overview Zoom Filter Details-on-Demand Browse Search Read Fact Read Comparison Read Pattern Manipulate Create	Delete Reorder Cluster Class Promote Average Abstract Instantiate Extract Compose Organize
--	--	---	--	---	--



Chapter 4 then looks in more detail at methods that dynamically focus on part of the space while maintaining a constant context, much like the visual system. Given the important role of text in knowledge crystallization, Chapter 5 focuses on methods for visualizing text. Chapter 6 is about visualization at other levels: infosphere, workspace, and visual object. Chapter 7 introduces some theory of information visualization. Finally, Chapter 8 discusses applications of information visualization and their implications.

Information visualization is a body of techniques that eventually will become part of the mainstream of computing applications just as computer graphics became part of the mainstream with the advent of bitmapped displays. At certain points, the development of technology crosses barriers of performance and cost that allow new sets of techniques to become widely used. This, in turn, has effects on the activities to which these techniques are applied. We believe this is about to happen with visualization technology and information visualization techniques. Information visualization is a

new upward step in the old game of using the resources of the external world to increase our ability to think. As Norman says,

*One method for expanding the power of the unaided mind is to provide external aids, especially notational systems, ways of representing an idea in some external medium so it can be maintained externally, free from the limits of working memory. (Norman, 1993, p. 246)*

Information visualization can help make us smart. Of course, leverage works both ways. It can also make us stupid by misguided mappings and unworkable user interfaces just as “chart junk” graphics makes information harder to comprehend. This set of readings is about efforts to puzzle out the difference between these two outcomes by invention and analysis. Not every idea in these papers is a good idea. But collectively they are part of the exploration of the space of possibilities for using visual computing to think.

# The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations

Ben Shneiderman  
Department of Computer Science,  
Human-Computer Interaction Laboratory, and Institute for Systems Research  
University of Maryland  
College Park, Maryland 20742 USA  
ben@cs.umd.edu

**Abstract:** A useful starting point for designing advanced graphical user interfaces is the Visual Information-Seeking Mantra: Overview first, zoom and filter, then details-on-demand. But this is only a starting point in trying to understand the rich and varied set of information visualizations that have been proposed in recent years. This paper offers a task by data type taxonomy with seven data types (1-, 2-, 3-dimensional data, temporal and multi-dimensional data, and tree and network data) and seven tasks (overview, zoom, filter, details-on-demand, relate, history, and extract).

*Everything points to the conclusion that the phrase 'the language of art' is more than a loose metaphor, that even to describe the visible world in images we need a developed system of schemata.*

E. H. Gombrich *Art and Illusion*, 1959 (p. 76)

## 1. Introduction

Information exploration should be a joyous experience, but many commentators talk of information overload and anxiety (Wurman, 1989). However, there is promising evidence that the next generation of digital libraries for structured databases, textual documents, and multimedia will enable convenient exploration of growing information spaces by a wider range of users. Visual language researchers and user-interface designers are inventing powerful information visualization methods, while offering smoother integration of technology with task.

The terminology swirl in this domain is especially colorful. The older terms of information retrieval (often applied to bibliographic and textual document systems) and database management (often applied to more structured relational database systems with orderly attributes and sort keys), are being pushed aside by newer notions of information gathering, seeking, or visualization and data mining, warehousing, or filtering. While distinctions are subtle, the common goals reach from finding a narrow set of items in a large collection that satisfy a well-

understood information need (known-item search) to developing an understanding of unexpected patterns within the collection (browse) (Marchionini, 1995).

Exploring information collections becomes increasingly difficult as the volume grows. A page of information is easy to explore, but when the information becomes the size of a book, or library, or even larger, it may be difficult to locate known items or to browse to gain an overview.

Designers are just discovering how to use the rapid and high resolution color displays to present large amounts of information in orderly and user-controlled ways. Perceptual psychologists, statisticians, and graphic designers (Bertin, 1983; Cleveland, 1993; Tufte, 1983, 1990) offer valuable guidance about presenting static information, but the opportunity for dynamic displays takes user interface designers well beyond current wisdom.

## 2. Visual Information Seeking Mantra

The success of direct-manipulation interfaces is indicative of the power of using computers in a more visual or graphic manner. A picture is often cited to be worth a thousand words and, for some (but not all) tasks, it is clear that a visual presentation—such as a map or photograph—is dramatically easier to use than is a textual description or a spoken report. As computer speed and display resolution increase, information visualization and graphical interfaces are likely to have an expanding role. If a map of the United States is displayed, then it should be possible to point rapidly at one of 1000 cities to get tourist information. Of course, a foreigner who knows a city's name (for example, New Orleans), but not its location, may do better with a scrolling alphabetical list. Visual displays become even more attractive to provide orientation or context, to enable selection of regions, and to provide dynamic feedback for identifying changes (for example, a weather map). Scientific visualization has the power to make atomic, cosmic, and common three-dimensional phenomena (such as heat conduction in engines, airflow over wings, or ozone holes) visible and comprehensible. Abstract information visualization has the power to reveal patterns, clusters, gaps, or

outliers in statistical data, stock-market trades, computer directories, or document collections.

Overall, the bandwidth of information presentation is potentially higher in the visual domain than for media reaching any of the other senses. Humans have remarkable perceptual abilities that are greatly under-utilized in current designs. Users can scan, recognize, and recall images rapidly, and can detect changes in size, color, shape, movement, or texture. They can point to a single pixel, even in a megapixel display, and can drag one object to another to perform an action. User interfaces have been largely text-oriented, so as visual approaches are explored, appealing new opportunities are emerging.

There are many visual design guidelines but the basic principle might be summarized as the Visual Information Seeking Mantra:

Overview first, zoom and filter, then details-on-demand  
Overview first, zoom and filter, then details-on-demand  
Overview first, zoom and filter, then details-on-demand  
Overview first, zoom and filter, then details-on-demand  
Overview first, zoom and filter, then details-on-demand  
Overview first, zoom and filter, then details-on-demand  
Overview first, zoom and filter, then details-on-demand  
Overview first, zoom and filter, then details-on-demand  
Overview first, zoom and filter, then details-on-demand

Each line represents one project in which I found myself rediscovering this principle and therefore wrote it down it as a reminder. It proved to be only a starting point in trying to characterize the multiple information-visualization innovations occurring at university, government, and industry research labs.

### 3. Task by Data Type Taxonomy

To sort out the prototypes and guide researchers to new opportunities, I propose a type by task taxonomy (TTT) of information visualizations. I assume that users are viewing collections of items, where items have multiple attributes. In all seven data types (1-, 2-, 3-dimensional data, temporal and multi-dimensional data, and tree and network data) the items have attributes and a basic search task is to select all items that satisfy values of a set of attributes. An example task would be finding all divisions in an organization structure that have a budget greater than \$500,000.

The data types are on the left side of the TTT characterize the task-domain information objects and are organized by the problems users are trying to solve. For example, in two-dimensional information such as maps, users are trying to grasp adjacency or navigate paths, whereas in tree-structured information users are trying to understand parent/child/sibling relationships. The tasks across the top of the TTT are task-domain information actions that users wish to perform.

The seven tasks are at a high level of abstraction. More tasks and refinements of these tasks would be

natural next steps in expanding this table. The seven tasks are:

**Overview:** Gain an overview of the entire collection.

**Zoom :** Zoom in on items of interest

**Filter:** filter out uninteresting items.

**Details-on-demand:** Select an item or group and get details when needed.

**Relate:** View relationships among items.

**History:** Keep a history of actions to support undo, replay, and progressive refinement.

**Extract:** Allow extraction of sub-collections and of the query parameters.

Further discussion of the tasks follows the descriptions of the seven data types:

**1-dimensional:** linear data types include textual documents, program source code, and alphabetical lists of names which are all organized in a sequential manner. Each item in the collection is a line of text containing a string of characters. Additional line attributes might be the date of last update or author name. Interface design issues include what fonts, color, size to use and what overview, scrolling, or selection methods can be used. User problems might be to find the number of items, see items having certain attributes (show only lines of a document that are section titles, lines of a program that were changed from the previous version, or people in a list who are older than 21 years), or see an item with all its attributes.

Examples: An early approach to dealing with large 1-dimensional data sets was the bifocal display which provided detailed information in the focus area and less information in the surrounding context area (Spence and Apperley, 1982). In their example, the selected issue of a scientific journal had details about each article, the older and newer issues of the journal were to the left and right on the bookshelf with decreasing space. Another effort to visualize 1-dimensional data showed the attribute values of each thousands of item in a fixed-sized space using a scrollbar-like display called value bars (Chimera, 1992). Even greater compressions were accomplished in compact displays of tens of thousands of lines of program source code (SeeSoft, Eick et al., 1992) or textual documents (Document Lens, Robertson and Mackinlay, 1993; Information mural, Jerding and Skasko, 1995).

**2-dimensional:** planar or map data include geographic maps, floorplans, or newspaper layouts. Each item in the collection covers some part of the total area and may be rectangular or not. Each item has task-domain attributes such as name, owner, value, etc. and interface-domain features such as size, color, opacity, etc. While many systems adopt a

multiple layer approach to dealing with map data, each layer is 2-dimensional. User problems are to find adjacent items, containment of one item by another, paths between items, and the basic tasks of counting, filtering, and details-on-demand.

Examples: Geographic Information Systems are a large research and commercial domain (Laurini and Thompson, 1992; Egenhofer and Richards, 1993) with numerous systems available. Information visualization researchers have used spatial displays of document collections (Korfhage, 1991; Hemmje et al., 1993; Wise et al., 1995) organized proximally by term co-occurrences.

**3-dimensional:** real-world objects such as molecules, the human body, and buildings have items with volume and some potentially complex relationship with other items. Computer-assisted design systems for architects, solid modelers, and mechanical engineers are built to handle complex 3-dimensional relationships. Users's tasks deal with adjacency plus above/below and inside/outside relationships, as well as the basic tasks. In 3-dimensional applications users must cope with understanding their position and orientation when viewing the objects, plus the serious problems of occlusion. Solutions to some of these problems are proposed in many prototypes with techniques such as overviews, landmarks, perspective, stereo display, transparency, and color coding.

Examples: Three-dimensional computer graphics and computer-assisted design are large topics, but information visualization efforts in three dimensions are still novel. Navigating high resolution images of the human body is the challenge in the National Library of Medicine's Visible Human project (North et al., 1996). Some applications have attempted to present 3-dimensional versions of trees (Robertson et al., 1993), networks (Fairchild et al., 1988), or elaborate desktops (Card et al., 1996).

**Temporal:** time lines are widely used and vital enough for medical records, project management, or historical presentations to create a data type that is separate from 1-dimensional data. The distinction in temporal data is that items have a start and finish time and that items may overlap. Frequent tasks include finding all events before, after, or during some time period or moment, plus the basic tasks.

Examples: Many project management tools exist, but novel visualizations of time include the perspective wall (Robertson et al., 1993) and LifeLines (Plaisant et al., 1996). LifeLines shows a youth history keyed to the needs of the Maryland Department of Juvenile Justice, but is intended to present medical patient histories as a compact overview with selectable items to get details-on-demand. Temporal data visualizations appear in

systems for editing video data or composing animations such as Macromedia Director.

**Multi-dimensional:** most relational and statistical databases are conveniently manipulated as multi-dimensional data in which items with  $n$  attributes become points in a  $n$ -dimensional space. The interface representation can be 2-dimensional scattergrams with each additional dimension controlled by a slider (Ahlberg and Shneiderman, 1994). Buttons can be used for attribute values when the cardinality is small, say less than ten. Tasks include finding patterns, clusters, correlations among pairs of variables, gaps, and outliers. Multi-dimensional data can be represented by a 3-dimensional scattergram but disorientation (especially if the user's point of view is inside the cluster of points) and occlusion (especially if close points are represented as being larger) can be problems. The technique of parallel coordinates is a clever innovation which makes some tasks easier, but takes practice for users to comprehend (Inselberg, 1985).

Examples: The early HomeFinder developed dynamic queries and sliders for user-controlled visualization of multi-dimensional data (Williamson and Shneiderman, 1992). The successor FilmFinder refined the techniques (Ahlberg and Shneiderman, 1994) for starfield displays (zoomable, color coded, user-controlled scattergrams), and laid the basis for the commercial product Spotfire (Ahlberg and Wistrand, 1995). Extrapolations include the Aggregate Manipulator (Goldstein and Roth, 1994), movable filters (Fishkin and Stone, 1995), and Selective Dynamic Manipulation (Chuah et al., 1995). Related works include VisDB for multidimensional database visualization (Keim and Kreigal, 1994), the spreadsheet-like Table Lens (Rao and Card, 1994) and the multiple linked histograms in the Influence Explorer (Tweedie et al., 1996).

**Tree:** hierarchies or tree structures are collections of items with each item having a link to one parent item (except the root). Items and the links between parent and child can have multiple attributes. The basic tasks can be applied to items and links, and tasks related to structural properties become interesting, for example, how many levels in the tree? or how many children does an item have? While it is possible to have similar items at leaves and internal nodes, it is also common to find different items at each level in a tree. Fixed level trees with all leaves equidistant from the root and fixed fanout trees with the same number of children for every parent are easier to deal with. High fanout (broad) and small fanout (deep) trees are important special cases. Interface representations of trees can use an outline style of indented labels used in tables of contents (Chimera and Shneiderman, 1993), a node and link diagram, or a treemap, in

which child items are rectangles nested inside parent rectangles.

Examples: Tree-structured data has long been displayed with indented outlines (Egan et al., 1989) or with connecting lines as in many computer-directory file managers. Attempts to show large tree structures as node and link diagrams in compact forms include the 3-dimensional cone and cam trees (Robertson et al., 1993; Carriere and Kazman, 1995), dynamic pruning in the TreeBrowser (Kumar et al., 1995), and the appealingly animated hyperbolic trees (Lamping et al., 1995). A novel space-filling mosaic approach shows an arbitrary sized tree in a fixed rectangular space (Shneiderman, 1992; Johnson and Shneiderman, 1991). The treemap approach was successfully applied to computer directories, sales data, business decision-making (Asahi et al., 1995), and web browsing (Mitchell et al., 1995; Mukherjea et al., 1995), but users take 10-20 minutes to accommodate to complex treemaps.

**Network:** sometimes relationships among items cannot be conveniently captured with a tree structure and it is useful to have items linked to an arbitrary number of other items. While many special cases of networks exist (acyclic, lattices, rooted vs. un-rooted, directed vs. undirected) it seems convenient to consider them all as one data type. In addition to the basic tasks applied to items and links, network users often want to know about shortest or least costly paths connecting two items or traversing the entire network. Interface representations include a node and link diagram, and a square matrix of the items with the value of a link attribute in the row and column representing a link.

Examples: Network visualization is an old but still imperfect art because of the complexity of relationships and user tasks. Commercial packages can handle small networks or simple strategies such as Netmap's layout of nodes on a circle with links criss-crossing the central area. An ambitious 3-dimensional approach was an impressive early accomplishment (Fairchild et al., 1988), and new interest in this topic has been spawned by attempts to visualize the World Wide Web (Andrews, 1995; Hendley et al., 1995).

These seven data types reflect are an abstraction of the reality. There are many variations on these themes (2 1/2 or 4-dimensional data, multitrees,...) and many prototypes use combinations of these data types. This taxonomy is useful only if it facilitates discussion and leads to useful discoveries. Some idea of missed opportunities emerges in looking at the tasks and data types in depth:

**Overview:** Gain an overview of the entire collection. Overview strategies include zoomed out views of

each data type to see the entire collection plus an adjoining detail view. The overview contains a movable field-of-view box to control the contents of the detail view, allowing zoom factors of 3 to 30. Replication of this strategy with intermediate views enables users to reach larger zoom factors. Another popular approach is the fisheye strategy (Furnas, 1986) which has been applied most commonly for network browsing (Sarkar and Brown, 1994; Bartram et al., 1995). The fisheye distortion magnifies one or more areas of the display, but zoom factors in prototypes are limited to about 5. Although query language facilities made it difficult to gain an overview of a collection, information visualization interfaces support some overview strategy, or should. Adequate overview strategies are a useful criteria to look for. Along with an overview plus detail (also called context plus focus) view there is a need for navigation tools to pan or scroll through the collection.

**Zoom:** Zoom in on items of interest. Users typically have an interest in some portion of a collection, and they need tools to enable them to control the zoom focus and the zoom factor. Smooth zooming helps users preserve their sense of position and context. Zooming could be on one dimension at a time by moving the zoombar controls or by adjusting the size of the field-of-view box. A very satisfying way to zoom in is by pointing to a location and issuing a zooming command, usually by clicking on a mouse button for as long as the user wishes (Bederson and Hollan, 1993). Zooming in one dimension has proven useful in starfield displays (Jog and Shneiderman, 1995).

**Filter:** filter out uninteresting items. Dynamic queries applied to the items in the collection is one of the key ideas in information visualization (Ahlberg et al., 1992; Williamson and Shneiderman, 1992). By allowing users to control the contents of the display, users can quickly focus on their interests by eliminating unwanted items. Sliders, buttons, or other control widgets coupled to rapid display update (less than 100 milliseconds) is the goal, even when there are tens of thousands of displayed items.

**Details-on-demand:** Select an item or group and get details when needed. Once a collection has been trimmed to a few dozen items it should be easy to browse the details about the group or individual items. The usual approach is to simply click on an item to get a pop-up window with values of each of the attributes. In Spotfire, the details-on-demand window can contain HTML text with links to further information.

**Relate:** View relationships among items. In the FilmFinder (Ahlberg and Shneiderman, 1994) users could select an attribute, such as the film's director, in the details-on-demand window and cause the director alphaslider to be reset to the director's name, thereby displaying only films by that director. Similarly, in SDM (Chuah et al., 1995), users can select an item and then highlight items with similar attributes or in LifeLines (Plaisant et al., 1996) users can click on a medication and see the related visit report, prescription, and lab test. Designing user interface actions to specify which relationship is to be manifested is still a challenge. The Influence Explorer (Tweedie et al., 1996) emphasizes exploration of relationships among attributes. and the Table Lens emphasizes finding correlations among pairs of numerical attributes (Rao and Card, 1994).

**History :** Keep a history of actions to support undo, replay, and progressive refinement. It is rare that a single user action produces the desired outcome. Information exploration is inherently a process with many steps, so keeping the history of actions and allowing users to retrace their steps is important. However, most prototypes fail to deal with this requirement. Maybe they are reflecting the current state of graphic user interfaces, but designers would be better to follow information retrieval systems which typically preserve the sequence of searches so that they can be combined or refined.

**Extract:** Allow extraction of sub-collections and of the query parameters. Once users have obtained the item or set of items they desire, it would be useful to be able to extract that set and save it to a file in a format that would facilitate other uses such as sending by email, printing, graphing, or insertion into a statistical or presentation package. An alternative to saving the set, they might want to save, send, or print the settings for the control widgets. Very few prototypes support this action, although Roth's recent work on Visage provides an elegant capability to extract sets of items and simply drag-and-drop them into the next application window.

The attraction of visual displays, when compared to textual displays, is that they make use of the remarkable human perceptual ability for visual information. Within visual displays, there are opportunities for showing relationships by proximity, by containment, by connected lines, or by color coding. Highlighting techniques (for example, bold-face text or brightening, inverse video, blinking, underscoring, or boxing) can be used to draw attention to certain items in a field of thousands of items. Pointing to a visual display can allow rapid selection, and feedback is apparent. The eye, the hand, and the

mind seem to work smoothly and rapidly as users perform actions on visual displays.

#### 4. Advanced Filtering

Users have highly varied needs for filtering features. The dynamic queries approach of adjusting numeric range sliders, alphasliders for names or categories, or buttons for small sets of categories is appealing to many users for many tasks (Shneiderman, 1994). Dynamic queries might be called *direct-manipulation queries*, since they share the same concepts of visual display of actions (the sliders or buttons) and objects (the query results in the task-domain display); the use of rapid, incremental, and reversible actions; and the immediate display of feedback (less than 100 msec). Additional benefits are no error messages and the encouragement of exploration.

Dynamic queries can reveal global properties as well as assist users in answering specific questions. As the database grows, it is more difficult to update the display fast enough, and specialized data structures or parallel computation are required.

The dynamic-query approach to the chemical table of elements was tested in an empirical comparison with a form-fill-in query interface. The counterbalanced-ordering within-subjects design with 18 chemistry students showed strong advantages for the dynamic queries in terms of faster performance and lower error rates (Ahlberg et al., 1991).

Dynamic queries usually permit OR combinations within an attribute with AND combination of attributes across attributes (conjunct of disjuncts). This is adequate for many situations since rapid multiple sequential queries allow users to satisfy their information needs. Commercial information-retrieval systems, such as DIALOG or Lexis/Nexis, permit complex *Boolean expressions* with parentheses, but widespread adoption has been inhibited by the difficulty of using them. Numerous proposals have been put forward to reduce the burden of specifying complex Boolean expressions (Reisner, 1988). Part of the confusion stems from informal English usage where a query such as List all employees who live in New York and Boston would result in an empty list because the "and" would be interpreted as an intersection; only employees who live in *both* cities would qualify! In English, "and" usually expands the options; in Boolean expressions, AND is used to narrow a set to the intersection of two others. Similarly, in the English "I'd like Russian or Italian salad dressing," the "or" is exclusive, indicating that you want one or the other but not both; in Boolean expressions, an OR is inclusive, and is used to expand a set.

The desire for *full Boolean expressions*, including nested parentheses and NOT operators, led us toward

novel metaphors for query specification. *Venn diagrams* (Michard, 1982), *decision tables* (Greene et al., 1990), and the innovative InfoCrystal (Spoerri, 1993) have been used, but these both become confusing as query complexity increases. We sought to support arbitrarily complex Boolean expressions with a graphical specification. Our approach was to apply the metaphor of water flowing from left to right through a series of pipes and filters, where each filter lets through only the appropriate documents, and the pipe layout indicates relationships of AND or OR. (Young and Shneiderman, 1993)

In this filter-flow model, ANDs are shown as a linear sequence of filters, suggesting the successive application of required criteria. As the flow passes through each filter, it is reduced, and the visual feedback shows a narrower bluish stream of water. ORs are shown two ways: within an attribute, multiple values can be selected in a single filter; and across multiple attributes, filters are arranged in parallel paths. When the parallel paths converge, the width of the flow reflects the size of the union of the document sets.

Negation was handled by a NOT operator that, when selected, inverts all currently selected items in a filter. For example, if California and Georgia were selected and then the NOT operator was chosen, those two states would become deselected and all the other states would become selected. Finally, clusters of filters and pipes can be made into a single labeled filter. This facility ensures that the full query can be shown on the display at once, and allows clusters to be saved in a library for later reuse.

We believe that this approach can help novices and intermittent users to specify complex Boolean expressions and to learn Boolean concepts. A usability study was conducted with 20 subjects with little experience using Boolean algebra. The prototype filter-flow interface showed statistically significant improved performance against a textual interface for comprehension and composition tasks. The filter-flow interface was preferred by all 20 subjects.

## 5. Summary

Novel graphical and direct-manipulation approaches to query formulation and information visualization are now possible. While research prototypes have typically dealt with only one data type (1-, 2-, 3-dimensional data, temporal and multi-dimensional data, and tree and network data), successful commercial products will have to accommodate several. These products will need to provide smooth integration with existing software and support the full task list: Overview, zoom, filter, details-on-demand, relate, history, and extract. These ideas are attractive because they present information rapidly and allow for rapid user-controlled exploration. If they are to be fully effective, some of these approaches require

novel data structures, high-resolution color displays, fast data retrieval, specialized data structures, parallel computation, and some user training.

Although the computer contributes to the information explosion, it is potentially the magic lens for finding, sorting, filtering, and presenting the relevant items. Search in complex structured documents, graphics, images, sound, or video presents grand opportunities for the design of user interfaces and search engines to find the needle in the haystack. The novel-information exploration tools—such as dynamic queries, treemaps, fisheye views, parallel coordinates, starfields, and perspective walls—are but a few of the inventions that will have to be tamed and validated.

**Acknowledgements:** This taxonomy was brewing in my mind at the time of the Gubbio, Italy conference on Advanced Visual Interfaces (May 1996). Stu Card's opening talk provoked me to start it in time for my closing talk, and I have refined it into the structure for the information visualization chapter in my forthcoming (1997) third edition of *Designing the User Interface*, Addison-Wesley Publishers. I am delighted and appreciative of Margaret Burnett and Wayne Citrin for giving me the chance to include and present these ideas in the Visual Languages 96 Conference.

## References

- Ahlberg, Christopher and Shneiderman, Ben, Visual information seeking: Tight coupling of dynamic query filters with starfield displays, *Proc. CHI94 Conference: Human Factors in Computing Systems*, ACM, New York, NY (1994), 313-321 + color plates.
- Ahlberg, Christopher and Shneiderman, Ben, AlphaSlider: A compact and rapid selector, *Proc. of ACM CHI94 Conference Human Factors in Computing Systems*, ACM, New York, NY (1994), 365-371.
- Ahlberg, Christopher, Williamson, Christopher, and Shneiderman, Ben, Dynamic queries for information exploration: An implementation and evaluation, *Proc. ACM CHI'92: Human Factors in Computing Systems*, ACM, New York, NY (1992), 619-626.
- Ahlberg, Christopher and Wistrand, Erik, IVEE: An information visualization & exploration environment, *Proc. IEEE Information Visualization '95*, IEEE Computer Press, Los Alamitos, CA (1995), 66-73.
- Andrew, Keith, Visualising cyberspace: Information visualisation in the Harmony internet browser, *Proc. IEEE Information Visualization '95*, IEEE Computer Press, Los Alamitos, CA (1995), 97-104.
- Asahi, T., Turo, D., and Shneiderman, B., Using treemaps to visualize the analytic hierarchy process, *Information Systems Research* 6, 4 (December 1995), 357-375.

- Bartram, Lyn, Ho, Albert, Dill, John, and Henigman, Frank, The continuous zoom: A constrained fisheye technique for viewing and navigating large information spaces, *Proc. User Interface Software and Technology '95*, ACM, New York, NY (1995), 207-215.
- Becker, Richard A., Eick, Stephen G., and Wilks, Allan R. Visualizing Network Data, *IEEE Transactions on Visualization and Computer Graphics 1*, 1 (March 1995), 16-28.
- Bederson, Ben B. and Hollan, James D., PAD++: A zooming graphical user interface for exploring alternate interface physics, *Proc. User Interfaces Software and Technology '94* (1994), 17-27.
- Bertin, Jacques, *Semiology of Graphics*, University of Wisconsin Press, Madison, WI (1983)
- Card, Stuart K., Robertson, George G., and York, William, The WebBook and the WebForager: An information workspace for the World-Wide Web, *Proc. CHI96 Conference: Human Factors in Computing Systems*, ACM, New York, NY (1996), 111-117.
- Carriere, Jeremy and Kazman, Rick, Interacting with huge hierarchies: Beyond cone trees, *Proc. IEEE Information Visualization '95*, IEEE Computer Press, Los Alamitos, CA (1995), 74-81.
- Chimera, Richard, Value bars: An information visualization and navigation tool for multiattribute listings, *Proc. CHI92 Conference: Human Factors in Computing Systems*, ACM, New York, NY (1992), 293-294.
- Chimera, Richard and Shneiderman, Ben, Evaluating three user interfaces for browsing tables of contents, *ACM Transactions on Information Systems 12*, 4 (October 1994).
- Chuah, Mei C., Roth, Steven F., Mattis, Joe, and Kolojechcik, John, SDM: Malleable Information Graphics, *Proc. IEEE Information Visualization '95*, IEEE Computer Press, Los Alamitos, CA (1995), 66-73.
- Cleveland, William, *Visualizing Data*, Hobart Press, Summit, NJ (1993).
- Egan, Dennis E., Remde, Joel R., Gomez, Louis M., Landauer, Thomas K., Eberhardt, Jennifer, and Lochbum, Carol C., Formative design-evaluation of SuperBook, *ACM Transactions on Information Systems 7*, 1 (January 1989), 30-57.
- Egenhofer, Max and Richards, J., Exploratory access to geographic data based on the map-overlay metaphor, *Journal of Visual Languages and Computing 4*, 2 (1993), 105-125.
- Eick, Stephen G., Steffen, Joseph L., and Sumner, Jr., Eric E., SeeSoft- A tool for visualizing line-oriented software statistics, *IEEE Transactions on Software Engineering 18*, 11 (1992) 957-968.
- Eick, Stephen G. and Wills, Graham J., Navigating Large Networks with Hierarchies, *Proc. IEEE Visualization '93 Conference*, (1993), 204--210.
- Fairchild, Kim M., Poltrock, Steven E., and Furnas, George W., SemNet: Three-dimensional representations of large knowledge bases, In Guindon, Raymonde (Editor), *Cognitive Science and its Applications for Human-Computer Interaction*, Lawrence Erlbaum, Hillsdale, NJ (1988), 201-233.
- Fishkin, Ken and Stone, Maureen C., Enhanced dynamic queries via movable filters, *Proc. CHI95 Conference: Human Factors in Computing Systems*, ACM, New York, NY (1995), 415-420.
- Furnas, George W., Generalized fisheye views, *Proc. CHI86 Conference: Human Factors in Computing Systems*, ACM, New York, NY (1986), 16-23.
- Goldstein, Jade and Roth, Steven F., Using aggregation and dynamic queries for exploring large data sets, *Proc. CHI95 Conference: Human Factors in Computing Systems*, ACM, New York, NY (1995), 23-29.
- Greene, S. L., Devlin, S. J., Cannata, P. E., and Gomez, L. M., No IFs, ANDs, or ORs: A study of database querying, *International Journal of Man-Machine Studies 32* (March 1990), 303-326.
- Hendley, R. J., Drew, N. S., Wood, A. S., Narcissus: Visualizing information, *Proc. IEEE Information Visualization '95*, IEEE Computer Press, Los Alamitos, CA (1995), 90-96.
- Humphrey, Susanne M. and Melloni, Biagio John, *Databases: A Primer for Retrieving Information by Computer*, Prentice-Hall, Englewood Cliffs, NJ (1986).
- Inselberg, Alfred, The plane with parallel coordinates, *The Visual Computer 1* (1985), 69-91.
- Jerding, Dean F. and Stasko, John T., The information mural: A technique for displaying and navigating large information spaces, *Proc. IEEE Information Visualization '95*, IEEE Computer Press, Los Alamitos, CA (1995), 43-50.
- Jog, Ninad and Shneiderman, Ben, Information visualization with smooth zooming on an starfield display, *Proc. IFIP Conf. Visual Databases 3*, Chapman and Hall, London (1995), 1-10.
- Johnson, Brian, and Shneiderman, Ben, Tree-maps: A space-filling approach to the visualization of hierarchical information structures, *Proc. IEEE Visualization '91*, IEEE, Piscataway, NJ (1991), 284-291.
- Keim, D. A. and Kriegel, H., VisDB: Database exploration using multidimensional visualization, *IEEE Computer Graphics and Applications* (September 1994), 40-49.
- Korfhage, Robert, To see or not to see -- Is that the query?, *Communications of the ACM 34* (1991), 134-141.
- Lamping, John, Rao, Ramana, and Pirolli, Peter, A focus + context technique based on hyperbolic geometry for visualizing large hierarchies, *Proc. of ACM CHI95 Conference: Human Factors in*



- Computing Systems*, ACM, New York, NY (1995), 401-408
- Laurini, R. and Thompson, D., *Fundamentals of Spatial Information Systems*, Academic Press, New York, NY (1992).
- Marchionini, Gary, *Information Seeking in Electronic Environments*, Cambridge University Press, UK (1995).
- Michard, A., A new database query language for non-professional users: Design principles and ergonomic evaluation, *Behavioral and Information Technology* 1, 3 (July–September 1982), 279–288.
- Mitchell, Richard, Day, David, and Hirschman, Lynette, Fishing for information on the internet, *Proc. IEEE Information Visualization '95*, IEEE Computer Press, Los Alamitos, CA (1995), 105-111.
- Mukherjea, Sougata, Foley, James D., and Hudson, Scott, Visualizing complex hypermedia networks through multiple hierarchical views, *Proc. of ACM CHI95 Conference: Human Factors in Computing Systems*, ACM, New York, NY (1995), 331-337 + color plate.
- North, Chris, Shneiderman, Ben, and Plaisant, Catherine, User controlled overviews of an image library: A case study of the Visible Human, *Proc. 1st ACM International Conference on Digital Libraries* (1996), 74-82.
- Pirolli, Peter, Schank, Patricia, Hearst, Marti, and Diehl, Christine, Scatter/gather browsing communicates the topic structure of a very large text collection, *Proc. of ACM CHI96 Conference*, ACM, New York, NY (1996), 213-220.
- Plaisant, Catherine, Rose, Anne, Milash, Brett, Widoff, Seth, and Shneiderman, Ben, LifeLines: Visualizing personal histories, *Proc. of ACM CHI96 Conference: Human Factors in Computing Systems*, ACM, New York, NY (1996), 221-227, 518.
- Rao, Ramana and Card, Stuart K., The Table Lens; Merging graphical and symbolic representations in an interactive focus + context visualization for tabular information, *Proc. CHI94 Conference: Human Factors in Computing Systems*, ACM, New York, NY (1994), 318-322.
- Reisner, Phyllis, Query languages. In Helander, Martin (Editor), *Handbook of Human-Computer Interaction*, North-Holland, Amsterdam, The Netherlands (1988), 257–280.
- Robertson, George G., Card, Stuart K., and Mackinlay, Jock D., Information visualization using 3-D interactive animation, *Communications of the ACM* 36, 4 (April 1993), 56-71.
- Robertson George G. and Mackinlay, Jock D., The document lens, *Proc. 1993 ACM User Interface Software and Technology*, ACM New York, NY (1993), 101-108.
- Sarkar, Manojit and Brown, Marc H., Graphical fisheye views, *Communications of the ACM* 37, 12 (July 1994), 73–84.
- Shneiderman, Ben, Tree visualization with tree-maps: A 2-d space-filling approach, *ACM Transactions on Graphics* 11, 1 (January 1992), 92-99.
- Shneiderman, Ben, Dynamic queries for visual information seeking, *IEEE Software* 11, 6 (1994), 70-77.
- Spence, Robert and Apperley, Mark, Data base navigation: An office environment for the professional, *Behaviour & Information Technology* 1, 1 (1982), 43-54.
- Spoerri, Anselm, InfoCrystal: A visual tool for information retrieval & management, *Proc. ACM Conf on Information and Knowledge Management* (1993).
- Tufte, Edward, *The Visual Display of Quantitative Information*, Graphics Press, Cheshire, CT (1983).
- Tufte, Edward, *Envisioning Information*, Graphics Press, Cheshire, CT (1990).
- Tweedie, Lisa, Spence, Robert, Dawkes, Huw, and Su, Hua, Externalising abstract mathematical models, *Proc. of ACM CHI96 Conference: Human Factors in Computing Systems*, ACM, New York, NY (1996), 406-412.
- Williamson, Christopher, and Shneiderman, Ben, The Dynamic HomeFinder: Evaluating dynamic queries in a real-estate information exploration system, *Proc. ACM SIGIR'92 Conference*, ACM, New York, NY (1992), 338-346. Reprinted in Shneiderman, B. (Editor), *Sparks of Innovation in Human-Computer Interaction*, Ablex Publishers, Norwood, NJ, (1993), 295-307.
- Wise, James A., Thomas, James, J., Pennock, Kelly, Lantrip, David, Pottier, Marc, Schur, Anne, and Crow, Vern, Visualizing the non-visual: Spatial analysis and interaction with information from text documents, *Proc. IEEE Information Visualization '95*, IEEE Computer Press, Los Alamitos, CA (1995), 51-58.
- Wurman, Richard Saul, *Information Anxiety*, Doubleday, New York (1989).
- Young, Degi and Shneiderman, Ben, A graphical filter/flow model for boolean queries: An implementation and experiment, *Journal of the American Society for Information Science* 44, 6 (July 1993), 327-339.

# Considering Visual Variables as a Basis for Information Visualisation

M.S.T. Carpendale  
Department of Computer Science  
University of Calgary

“Communication is too often taken for granted when it should be taken to pieces.” (Fiske’91)

## Abstract

The purpose of this report is to discuss Bertin’s concepts about visual variables [Bertin 67/83], which are no longer readily available since they are out of print. Bertin introduced the idea of visual variables as part of his extensive analysis of cartography in the creation of what he terms data graphics as part of his *Semiology of Graphics* [Bertin 67/83]. In this report these concepts are discussed in terms of computational information visualisation instead of printed cartography. The discussion will centre on how these visual variables can be used in the creation of visual representations for the purpose of information visualisation.

## 1. Introduction

Information can come in many forms from the concrete to the conceptual and the abstract. If this information is to be capable of communicating to or informing people, it must be represented in a manner that is understandable. This creation of a visual representation from the information is a significant part in the process of creating an information visualisation. Since the information that is being represented may not have any obvious visual manifestation this process of creating mapping from the information to the visual representation can be non-trivial. In approaching this problem there are sources from many fields of study that have the potential of providing useful advice. These include cognitive science (Ware 2000), information design (Tuft 1983, 1987, 1990), linguistics (Horn 2000), comics (McCloud 1993 2000), film (Dondis) and cartography (Bertin 1967/83).

## 2. The Importance of Representation

Creating or changing representations in general, or visual representations in particular, is very significant in regards to issues of comprehension and interpretability of the information that is represented.

For this discussion the term representation is used as defined by Marr (1982) to mean a formal system by which the information or data can be specified. Defined in this way a given representation provides specific information about the data and differing representations may more readily reveal differing aspects of the data. To explain the significance of changes in representation Marr’s example is paralleled. Arabic, Roman and binary representations can be provided for the concept of the number thirty-four, giving 34, XXXIV, and 100010 respectively. In this example, Arabic numerals reveal information about powers of ten while binary representations reveal information about powers of two. The information about powers of ten is available in all three of these representations, however the degree of accessibility varies considerably. This indicates that choice of representation is of fundamental importance if the information or data that we start with is to be capable of informing others.

The visualisation process involves several representational mappings. Information or data has initially been observed, gathered or generated in some manner and perhaps stored for later use. This in itself involves the creation of a representation. Each time a representation is created process of abstraction has been used. The act of abstracting is taken to mean to summarise, or to state the essence of the information concerned. Most abstractions are either more or less than direct transformations in that the new representation was arrived at through more than simply change of form without alteration of quantity or value of the information. That is, in creating a new representation certain aspects of the information are brought to the fore and others are perhaps obscured or even omitted. It is this process of abstraction in the creation of a new representation that selects which aspects of the data are to be the most accessible. In creating visualisations the first information or data representation will not often have a visual form; therefore a second process of abstraction may be required to create a visual representation.

In a given representation, information may be present but hard to find. Useful representations allow people to find relevant information and allow people to compute desired conclusions. Computations may be difficult or “for free” depending on representations.

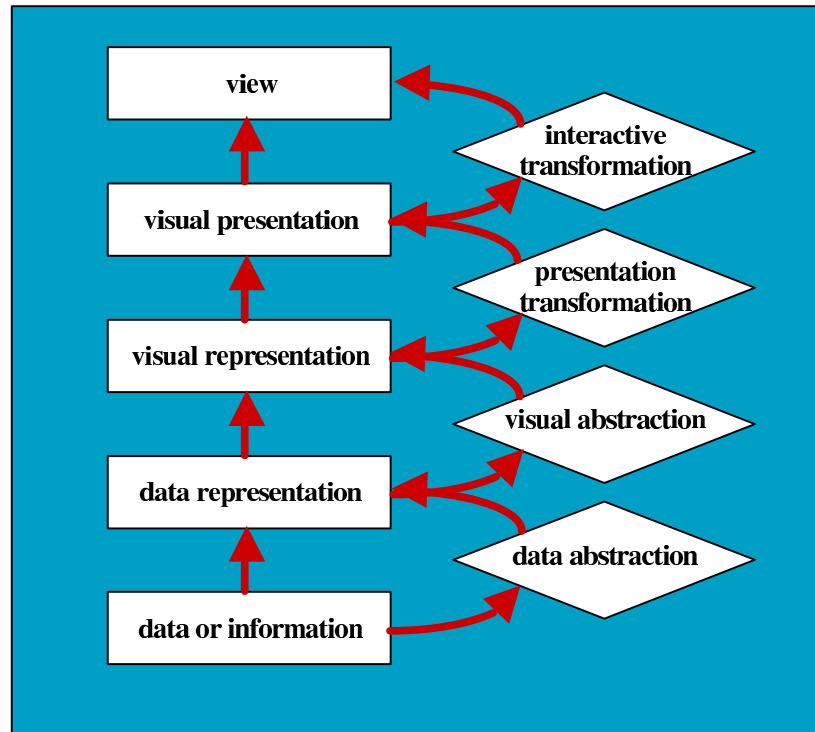


Figure 1: Visualisation pipeline

Choice of representation enables or complicates performance of particular tasks, there are several changes of representation involved in developing a visualisation and that each one of these usually involves a process of abstraction. And the each abstraction effects what information is available for use. The goal of this discussion is to dissect the process, by which a visual representation can be created, increasing consciousness of the process of representational transformations and abstractions. In turn this increased awareness may help in the creation of representations that more appropriately match their tasks.

### 3. Getting Started

#### 3.1 Basic Units

This is a practical look at how to go about creating a visual mapping that is capable of communicating. Our most familiar mode of communication is with words. Words are composed of sounds or phonemes and these phonemes are constructed from the letters, usually one or two letters make a phoneme. Note that the letters are meaningless in themselves, as are the phonemes [Sausure]. It is not until phonemes are grouped together to form words that they have meaning.

The first question in this discussion is: are there corresponding basic visual units? There is still considerable debate on this subject with the majority opinion being negative. However; Jacques Bertin [1967/83] developed a practical approach for his data graphics that is based on the opinion that there is a basic visual unit and that there are a describable ways of changing this basic unit. What then is the most basic visual unit one make on a blank piece of paper? One can make a mark.

Bertin [1967/83] defines a mark as something that is visible and can be used in cartography to show relationships within sets of data. He names the different ways that a mark can be varied as visual variables. The next question is how can one vary this mark in a manner that is meaningful? Marks can be varied by where we place them on the page and by their visual characteristics such as size, shape, value, orientation, colour and texture.

### 3.2 Disclaimer

Bertin prefaced his work with the following disclaimer. He states that in developing this approach he was considering the creation of data graphics, on white paper, that are printable with readily available means. That he wanted the data to be visible at a glance and that he is considering normal book reading conditions. This includes normal and constant lighting, and reading distance of up to an arm's length. It is important to also preface this report with such a disclaimer.

One of the principle reasons for writing this report is to make Bertin's concept of visual variables, which is unfortunately now out of print, available to students of information visualisation. However, this is not simply a reiteration of Bertin's concepts, there are several notable differences. Here, these concepts are discussed as they pertain to information visualisations that are created for and presented on computational displays. Also, Bertin classifies visual variables by what he terms perceptual level of organisation. Applying Bertin notions to information visualisation has led to re-phrasing the classification in terms of visual interpretation tasks. One reason for this shift is that it supports a practical emphasis and application. Another reason is that in the approximately thirty years since Bertin first published, research into perceptual capabilities has advanced considerably and has also been considered in terms of information visualisation [Ware90]. This consideration in terms of visual interpretation tasks has led to some variance in the concepts, particularly the definition of the associative interpretation tasks

As the computer emerges as a medium in its own right, it is important to improve our understanding of the capabilities of computational presentation space. For example, before using a new tool, even one as simple as a pencil or a brush, an artist will test it to gain knowledge of the characteristic range of marks that can be made. Just as an artist benefits from knowledge of the tools they are using, a person creating an information visualisation for a computer will benefit from fuller understanding of the possible representation choices. However, this is not meant to be definitive description of computational visual representation space but is merely a practical first pass. In comparison to printing computational display is a relatively new medium. Readily available computational displays vary considerably from those of printing and, what is more, are in a state of flux. For example, a typical computational display is backlit and has poor comparatively poor resolution (currently up to 1600 x 1200) in contrast to printing. Also new physical displays are arising on a regular basis as well as new software capabilities. These will continue to change what it is possible to display visually. For instance on a computational display one can consider; movement (speed, frequency, onset, style), the quasi 3D display (depth, occlusion, aerial perspective, binocular disparity, stereo viewing), illumination, transparency and three readily available colour channels (either red, green and blue or hue, saturation and value).

While Bertin's basic idea translates well from data graphics to computational visualisation, there are several distinctions and in all probability more distinctions or at least more precise characterisation of these distinctions will develop with use.

### 3.3 Marks

A mark can be a point, a line, an area and, on a computational display, a surface and a volume. Once a mark has been made and that mark is used to represent something other than itself it is frequently referred to as a sign.

#### Points

Points theoretically have no size. Symbolically they represent a mathematically dimensionless location. That is, a point represents the concept of location (x, y, z) independently from the size and shape that it manifests. Points of course will have such things as size, shape and colour in order that they can be perceived but they still symbolically stand for a dimensionless point. Points operate in a 1D, 2D, or 3D space. Marks that indicate points can vary in all visual variables. "A point represents a location on the plane that has no theoretical length or area. This signification is independent of the size and character of the mark which renders it visible." [page 44 Bertin 67/83]

#### Lines

Lines have length but no theoretical width. Symbolically they can represent such things as a boundary, a connection, a separation and an edge. Similarly to points, lines manifest visually with some thickness in order that they may be seen. But they stand for the location

of the line. A change in the visual characteristics of the line such as, thickness (size), texture, or colour does not change the meaning of the line. Changing its location will change its meaning. Lines operate in a 2D or 3D space. "A line signifies a phenomenon on the plane which has measurable length but no area. This signification is independent of the width and characteristic of the mark which renders it visible." [page 44 Bertin 67/83]

### **Areas**

An area on the other hand has length and width. Changing these properties changes the meaning of the area. For instance, if an area represents a country, decreasing the size of the area on the map would signify decreasing the size of the country. Areas operate in a 2D space. An area can change in position, colour, value, or texture but not in size, shape or orientation without making the area itself have a different meaning. "An area signifies something ... [in the presentation] that has measurable size. This signification applies to the entire area covered by the visible mark." [page 44 Bertin 67/83]

### **Surfaces**

Surfaces are similar to areas in that they have length and breadth but they exist in a 3D space and have no theoretical thickness. They can represent such things as connections, volume separations and volume edges. A surface can change in colour, value, or texture and size in thickness only without changing its meaning. Other changes in position, size, shape or orientation will give the surface a different meaning. A plane is a flat surface.

### **Volumes**

Volumes have length, width and depth. Their size is their meaning. They exist in a 3D space. A volume can change in position, colour, value, or texture but not in size, shape or orientation without making the volume itself have a different meaning. Similarly to an area, a volume signifies something that has measurable size. This signification applies to the entire volume covered by the visible mark.

## **4. The Visual Variables**







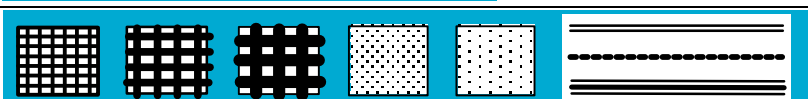
In this section I will introduce Bertin's visual variables, discuss briefly some differences caused by computational display and augment them minimally according to these differences. These are briefly introduced and illustrated in Table 1.

In computational presentation the addition of **motion** as a visual variable is important. Changing a marks motion is a new visual variable available for computational presentation. There are many changes on motion that are possible. These include direction, speed, frequency, rhythm, flicker, trails, and style. For further discussion of the use of motion in information visualisation see Bartram [1998].

## **5. Characteristics of Visual Variables**

Considerable power in creating different visual representations comes from choosing which visual variable would be most appropriate to represent each aspect of the information. The ability to make these choices can be greatly enhanced by understanding how a change in a particular visual variable is likely to affect the performance of a particular task. This is Bertin's list of visual variable characteristics.

The first four, selective, associative, quantitative and order, are visual interpretation tasks. They allow us to classify visual variables according to whether changes in a given variable enable the performance of these different types of visual interpretation tasks. The last characteristic in this list, length, addresses the issue of how many changes in a particular visual variable can be used effectively.

Bertin's Original Visual Variables	
<b>Position</b> changes in the x, y location	
<b>Size</b> change in length, area or repetition	
<b>Shape</b> infinite number of shapes	
<b>Value</b> changes from light to dark	
<b>Colour</b> changes in hue at a given value	
<b>Orientation</b> changes in alignment	
<b>Texture</b> variation in 'grain'	

**Table 1: These are Bertin's visual variables**

- 5.1 Selective.** A visual variable is said to be selective if a mark changed in this variable alone makes it easier to select that changed mark from all the other marks. This task is about the selection of an individual mark as distinct from other marks. The question to be asked is: Is change in this visual variable alone enough to allow us to select it from a group?
- 5.2 Associative.** A visual variable is said to be associative if marks that are like in other ways can be grouped according to a change in this visual variable. This means that several marks can be grouped across changes in other visual variables. For example all yellow marks can be thought of as a group even if they are in different locations or have different shapes. The question to be asked is: Is a change in this visual variable enough to allow us to perceive them as a group? Bertin uses the term associative differently (see page 48 in [Bertin 67/83]).
- 5.3 Quantitative.** A visual variable is said to be quantitative if the relationship between two marks differing in this visual variable can be seen as numerical. For instance, one line can be seen as being four times as long as another line. These are not necessarily precise numerical readings but are often read as ratios of one mark to another. For instance, one line can be seen as being four times as long as another line. The question to be asked is: Is there a numerical reading obtainable from changes in this visual variable?
- 5.4 Order.** A visual variable is said to be ordered if changes in this visual variable support ordered readings. That is a change in an ordered visual variable will automatically be read as either more or less. For instance a change in value will be seen as either less dark or more light. The question to be asked is: Are changes in this variable perceived as ordered?
- 5.5 Length.** Length is a slightly different kind of characteristic. The length of a visual variable is the number of changes that can be used and still retain the task supporting characteristics that are usually associated with this visual variable. For, example how many changes in value (shades of grey) can still be recognised with confidence as separate. The question to be asked is: across how many changes in this visual variable are distinctions recognisable?

## 6. Discussing the Characteristics of Each Visual Variable

### 6.1 Position

A mark on a printed page changes in position when it is moved up or down or to the left or to the right. In print, position changes are changes in the x, y location of the mark. On a computational display, if the representation space is 2D this remains the same, if it is 3D, there are three positional variables, x, y, and z.

While printed graphics are 2D, computational visualisations are created in a computer's quasi 3D-display space. The displays commonly in use at this point are neither completely 2D nor actually 3D. They are 2D in that they present everything on a flat 2D surface. They are 3D in that we can use mathematics to enable us to create displays that appear 3D but in fact they are still always projected onto a 2D screen. The amount of 3D effect can vary from very little as in windows that both exist in multiple over-lapping layers and still seem to be all on the same surface, to as 3D as immerse and stereo viewing can support.


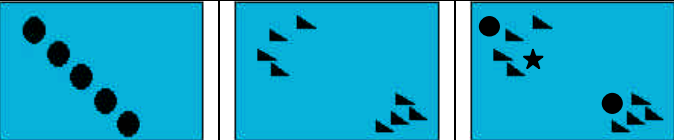
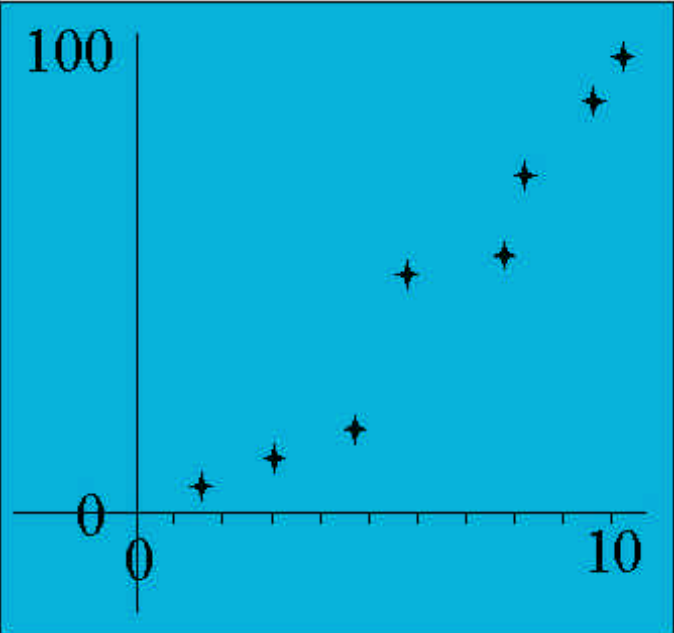
Visual Variable: Position		
✓	selective	
✓	associative	
✓	quantitative	
✓	order	
✓	length	
✓		

Table 2: Visual variable position and interpretation tasks

**Position: Selective.** Marks that are the same in all respects except for position can easily being distinguished and interpreted as different. In Table , row 1, columns 3, 4 and 5 show, respectively, circular, linear and triangular marks that differ only in position. They are all selectable based on viewing their location. For instance, if asked to select the lower most

mark from Table 2, row 1, column 5, you can do so readily by simply looking at it. A change in this visual variable alone is sufficient to allow us to select it from a group.

**Position: Associative.** Changes in positional visual variables can be used to create groups of marks that can be interpreted as belonging together. Table 2, row 2 shows three examples of this. Table 2, row 2, column 3 shows a diagonal row of circles that are clearly positioned to indicate that they are to be considered as a group. Table 2, row 2, column 4 shows distinct groups of triangles. Here the use of position visually separates these marks into two groups in spite of the fact that all that marks are the same shape and colour. Table 2, row 2, column 5 also shows two groups. In these two groups some of the marks have differing shapes however, the use of position clearly groups them. If asked for the group in the top left corner one can readily see which marks to include. This illustrates that several marks can be grouped by position across changes in other visual variables.

**Position: Quantitative.** Position is frequently used to indicate a numerical value. The graph in Table 2 that spans rows 3, 4 and 5 and columns 3, 4 and 5 illustrates this. Depending on the care taken with the visual presentation and the incorporation of viewing cues relatively precise numerical readings are obtainable. For example, the small marks on the x-axis greatly aid the ease of retrieving a numerical reading in the x direction. However, numerical readings in the y direction are also obtainable. Definitely the relationship between two marks differing in position can be interpreted as numerical.

**Position: Order.** Changes in position are readily orderable. This holds true for both up-down interpretations and left-right interpretations. The ordered interpretation of all the marks in the graph in Table 2 rows 3, 4 and 5 and columns 3, 4 and 5 will be consistent.

**Position: Length.** The positional variables have exceptional length in that a considerable number of changes in position are still perceived as distinct and therefore will still retain the task supporting characteristics that are usually associated with position. While theoretically infinite in that one can theoretically always place a new mark in between two existing marks, practically in computational presentation space the length of the positional visual variables is dependent on the resolution of the display.

Position is the most versatile and most powerful visual variable and, fortunately, there is positional visual variable for each dimension of the presentation. However, while both of the planar positional variables fully support all of the four of the visual interpretation tasks, this does not seem to be entirely the case for the three positional variables in 3D presentations. For more complete discussion of this issue see [Ware97; Ware00].

## 6.2 Size




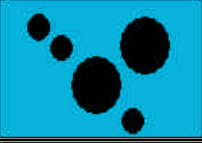

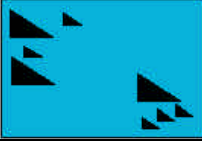
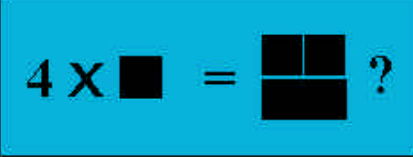
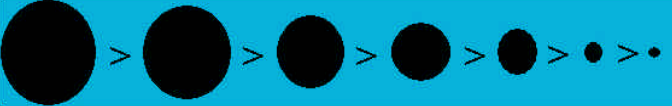
Changing a mark or sign's size is achieved by changes in length, area, volume or by repetition of a number of equal signs. (Table 1, row 2). Only points, lines or planes can be changed in size without causing a change in the interpretation of the sign. A change in size is a change in the dimensions of the sign. For instance, a sign that represents a point in the presentation can be made larger according to needs for visibility while still representing that point.

**Size: Selective.** A change in size is selective since if a sign is changed in size alone it will become distinct and therefore selectable from the other signs. In Table 1, row 1, it is a simple interpretation task to select either the smaller or the larger sign. Signs that are the same in all respects except for size can easily be distinguished and interpreted as different.

**Size: Associative.** Changes in size can be used to create groups of signs that can be interpreted as belonging together. Table 3 row 2 shows three examples of this. Here the use of size visually separates these signs into two groups, a group of smaller signs and a group of larger signs. A change in size alone is sufficient to allow us to perceive signs of similar size as belonging to a group.



**Size: Quantitative.** Size can sometimes be used to indicate a numerical value. However, numerical readings interpreted from changes in size alone are usually approximate and often less the ideally accurate. Table row 3 illustrates this. If the change in size is achieved through repetition of like marks, numerical readings are more readily obtainable. However, if changes in size are achieved by changes in area or volume, they are much more difficult to interpret. A size change that is achieved by a change in one dimension only can be more amenable to comparative numerical interpretations. Use of changes in size when the intention is to make information about comparative numerical values readily available can be problematical. Therefore this use of the visual variable size should be done with caution.

Visual Variable: Size		
✓	selective	  
✓	associative	  
≈	quantitative	
✓	order	
✓	Length	<ul style="list-style-type: none"> <li>• theoretically infinite but practically limited</li> <li>• association and selection ~ 5 and distinction ~ 20</li> </ul>

**Table 3: Visual variable size and interpretation tasks**


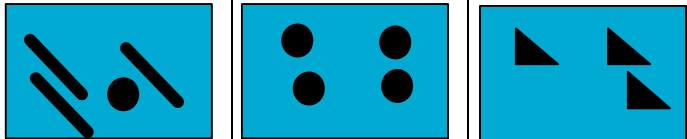







**Size: Order.** Changes in size are readily orderable. In fact words such as larger or smaller that express changes in size are among the most common ways of indicating order verbally. This is probably due to the ease with which order can be interpreted from changes in size. Table 3 row 4 illustrates this. The interpretation of order achieved through changes in size is consistent across all readings. That is a change in size will automatically be read as either more or less.

**Size: Length.** While changes in size are theoretically infinite in that one can theoretically always make a minimally larger size, like position this is also limited by the resolution of the display. However, the length of the visual variable size also suffers from a much more practical limitation. To achieve the signification desired, a change in size must be interpretable as change in size. That is the two sizes must be visually distinct. Practically, while it is possible for us to interpret quite small changes in size when they are immediately adjacent, much larger changes in size are needed if it is to remain interpretable across greater distances. Therefore while several changes in size, perhaps in the range of forty to fifty, are usable if they are placed adjacently, only comparatively few, perhaps as limited as in the range of approximately five, are usable if they are to be placed separately in the display.

### 6.3 Shape

Changing a mark's shape is achieved by any change in outline that does not include a change in size. (Table 1, row 3).

**Shape: Selective.** A change in shape can be selective. For example, in Table 4 row 1, it is possible to select the circle in column 3, the triangle in column 4 and the square in column 5 as being distinct. However, the simplicity of this interpretation task starts to break as the number and proximity of other signs start to increase. Examine Figure 2. It is not a simple interpretation task to select the shape in the small image on the left from the group of shapes in the image on the right. There are five of them. While changes in shapes are distinguishable, this distinction can often require considerable interpretation effort. Therefore I consider shape as partially selective.

Visual Variable: Shape		
	selective	
	associative	
	quantitative	
	order	
	length	 theoretically infinite

**Table 4: Visual variable shape and interpretation tasks**

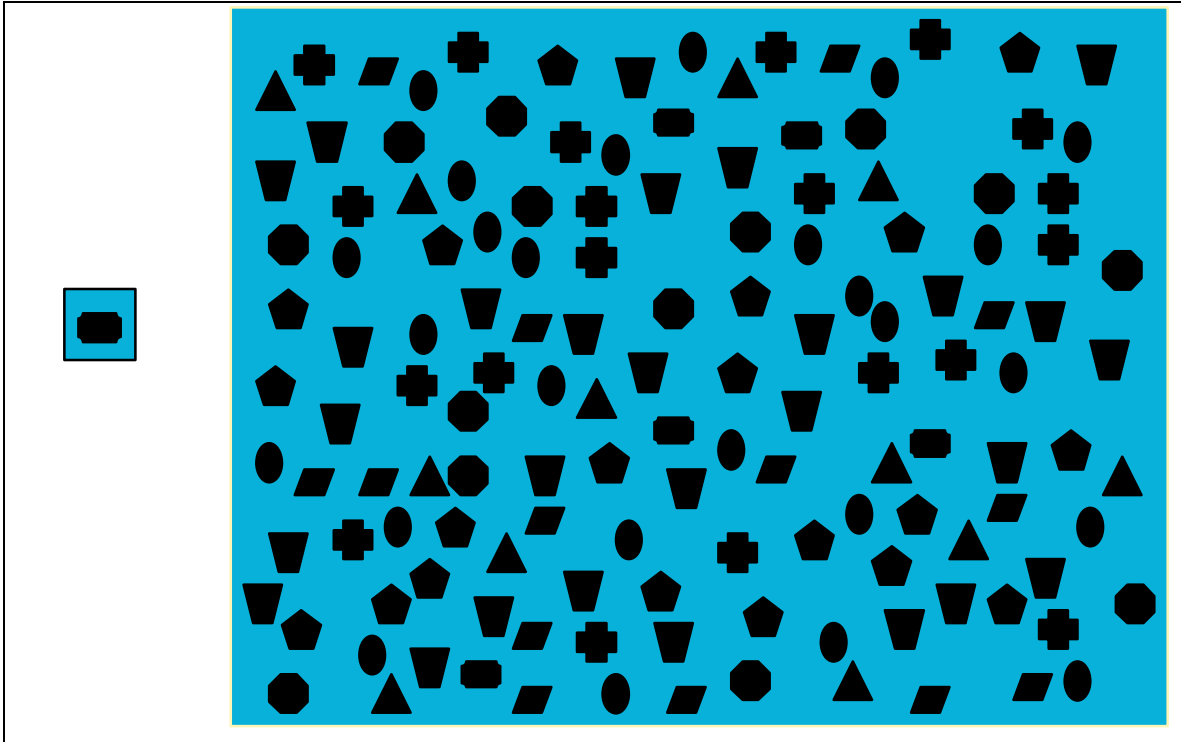
**Shape: Associative.** The question is can a change in shape alone be enough to allow us to perceive all the signs with this shape as a group? In Table 4 row 2 there are several examples of small groupings with simple changes in shape. In these small examples, shape is associative. That is, one can interpret the circles in Table row 2 column 4 as a group in spite of the fact that they are not grouped positionally. However, once again if one examines Figure 2, even once one has discovered all instances of a particular shape, it is difficult to interpret them as a group. This is not to say that if, for instance, all the crosses represent the presence of a mine of a map, one can not find another cross and interpret it as signifying that there is a mine at this location. This is a symbolic interpretation. However, a quick visual interpretation of all the mines is not very accessible. Therefore I consider shape as partially associative.

**Shape: Quantitative.** A change in shape, that is not accompanied by a change in size, does not readily provide a numerical interpretation. Therefore shape is not a quantitative visual variable.

**Shape: Order.** Signs that change in shape only do not support ordered readings. If they have even approximately the same area, a circle will not be interpreted as more or less than

parallelogram. If pressed to order a set of shapes people will order them different according to criteria of their own such as personal preference. Shape is not an ordered visual variable.

**Shape: Length.** The shape of a sign of a particular area can be varied infinitely.



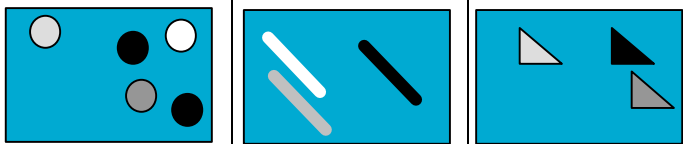
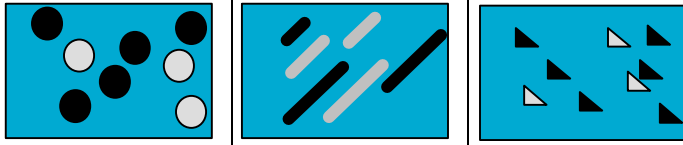
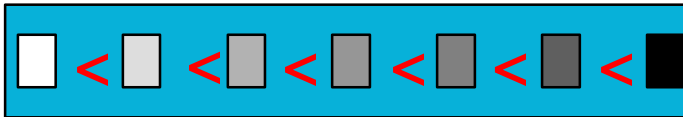
**Figure 2:** Try the interpretation tasks of selecting the particular shape that is placed on the left hand side and then of associating all examples of this shape as a group

The representational power of shape comes from its infinite length and from symbolic interpretation. A shape can be associated with a meaning and become a sign for that meaning. However, for symbolic meaning to be effective the link between the shape and the intended meaning must be explicit. This symbolic link can be cultural and our alphabets are an excellent example of this. Or this symbolic link can be stated at the time of use for example, in a legend with a map.

#### 6.4 Value

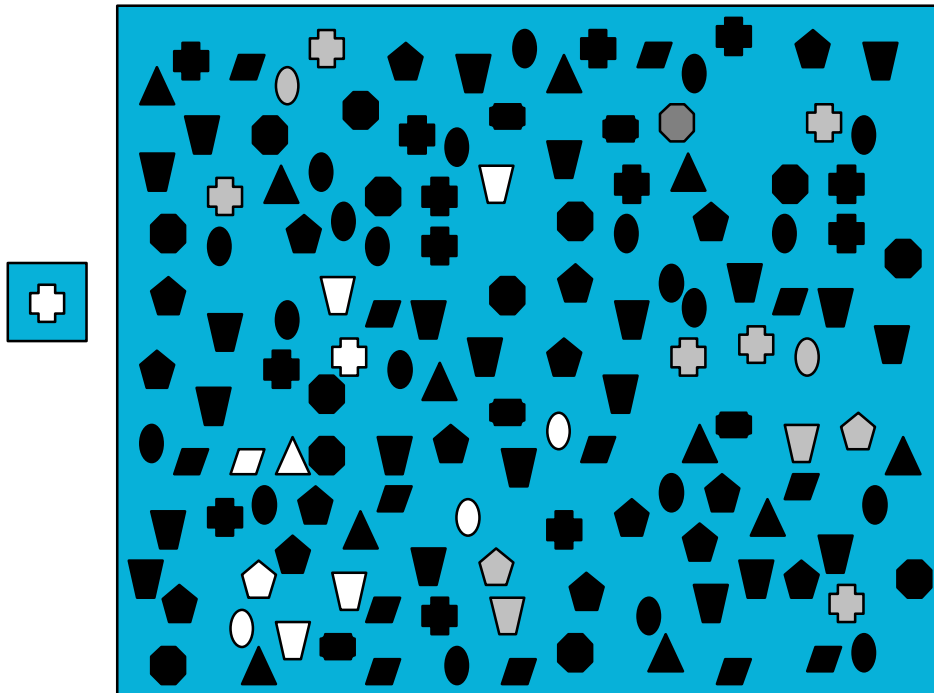
Changing a mark's value is achieved by changes in darkness or lightness of the mark (Table 1, row 4). This provides a range of shades of grey and is considered as independent of changes in colour. That is the value of a mark will be taken as its relative lightness or darkness alone. The extremes in value, white and black can be approached through changes in any hue, such as red or blue. This use of the term value is consistent with its use in the increasingly common HSV (hue, saturation, and value) breakdown of changes in colour space.

**Value: Selective.** Marks that change in value can easily being distinguished and interpreted as different. In Table , row 1, columns 3, 4 and 5 show, respectively, circular, linear and triangular marks that differ value. They are all selectable based on viewing their value. For instance, if asked to select the grey circle from Table 5 row1, column 3, you can do so readily by simply looking at it. A change in value alone is sufficient to allow us to select it. Note in Figure 3 how value remains selectable when the numbers of surrounding signs increases. It is, for instance, quite possible to select the dark grey sign.

Visual Variable: Value		
✓	selective	
✓	associative	
≠	quantitative	
✓	order	
✓	length	<ul style="list-style-type: none"> <li>• theoretically infinite but practically limited</li> <li>• association and selection ~ &lt; 7 and distinction ~ 10</li> </ul>

**Table 5: Visual variable value and interpretation tasks**

**Value: Associative.** Changes in value are associative since signs that are like in other ways can be grouped according to a change in value (see Table 5, row 2). Also value can be used to group several marks across changes in other visual variables. Note how in Figure 3 this holds true as the number of surrounding signs increases. For example, associating all white signs is readily accessible interpretation task.



**Figure 3: Try the interpretation tasks of selecting a particular value or associating all examples of a particular value as a group**

**Value: Quantitative.** Changes in value do not provide numerical readings since the relationship between two signs differing in value only, is not seen as numerical. For instance, while one grey may be seen as darker or lighter than another grey it will not be seen as say four times as dark as the other grey. Even ratios between shades of grey are not easily visually interpretable. The visual variable value is not quantitative.

**Value: Order.** Value is ordered since changes in value support ordered readings. That is a change in value will automatically be read as either more or less than the previous value. In fact, changes in value are very powerfully ordered. Significantly so that they will override reading from changes in other visual variables. See the discussion under colour about this.

**Value: Length.** The number of changes possible in value is theoretically infinite but is practically limited. To retain the facility it provides for selective and associative interpretation tasks it is advisable to limit the length of this variable to six or seven changes that include black and white. If the required reading is between adjacent signs it may be possible to increase this length slightly in that we can interpret changes in shades of grey more readily when there are no visual gaps between.

## 6.5 Colour

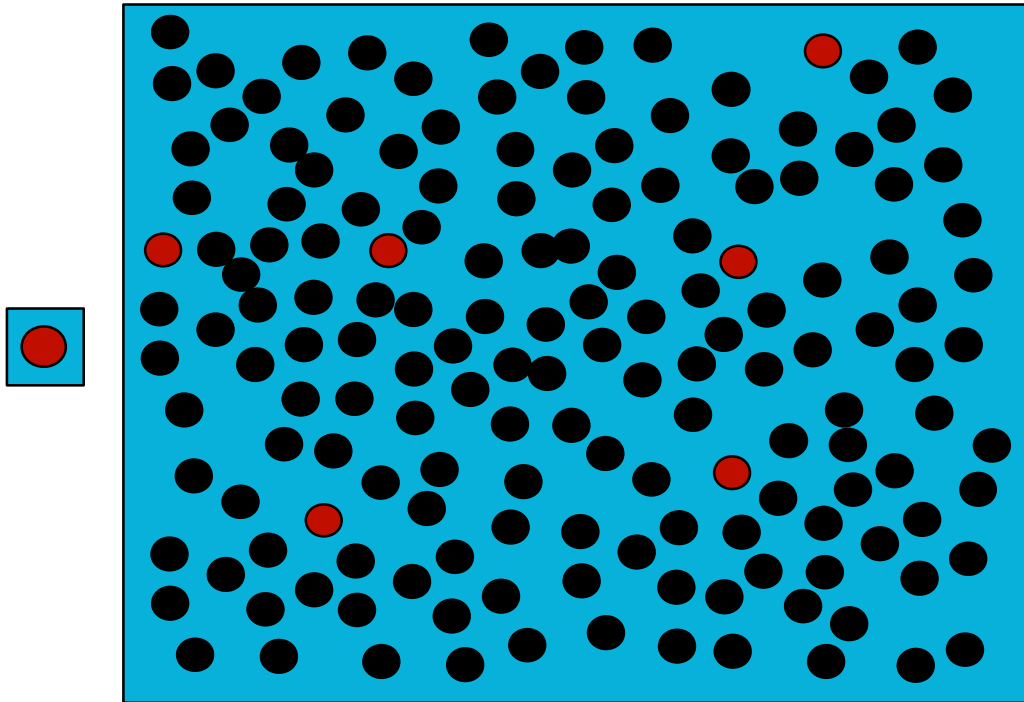
Changing a mark's colour in Bertin's [67/83] terms involves changes in hue without changes in value (Table 1, row 5). On a computational display there are other readily available changes that can loosely be termed changes in colour. These include saturation and transparency. This discussion considers variations in colour that can be achieved at a constant value.

Visual Variable: Colour		
✓	selective	
✓	associative	
≠	quantitative	
≠	order	
✓	length	<ul style="list-style-type: none"> <li>• theoretically infinite but practically limited</li> <li>• association and selection ~ &lt; 7 and distinction ~ 10</li> </ul>

**Table 6: Visual variable colour and interpretation tasks**

**Colour: Selective.** Colour is selective since a mark changed in colour alone makes it easy to select from all the other marks. Table 6 row 1 illustrates how colour is selectable. Figure 4 shows how this selectivity does not break down as the number of marks increases. Colour appears to be particularly selectable for pure and saturated hues.

**Colour: Associative.** Colour is associative since marks that are like in other ways can be grouped according to a change in colour. Also marks of one colour can be grouped across changes in other visual variables. For example all red triangles in Table 6a row 3 can be thought of as group even if they are in different locations.



**Figure 4:** Try the interpretation task of associating all examples of a red as a group

**Colour: Quantitative.** Colour is not quantitative since the relationship between two marks differing in colour will not be read numerically. For instance, red is not seen as being four times as coloured as blue. There is no numerical reading obtainable from changes in colour.

**Colour: Order.** Colour is not ordered since changes in colour do not easily lend themselves to readings of greater or lesser. For instance changing a colour from red to green will not be seen having either less colour or more colour. However, it is often said that the rainbow provides an ordered reading for colour. This ordering does not hold up in printed graphics and therefore seems unwise to use in computational visualisation. See the discussion below on colour ordering and the rainbow scale

**Colour: Length.** Similarly to value, the number of changes possible in colour is theoretically infinite but is practically limited. To retain the facility that it provides for selective and associative interpretation tasks it is advisable to limit the length of this variable to six or seven changes. If the required reading is between adjacent signs it may be possible to increase this length slightly in that we can interpret changes hue more readily when there are no visual gaps between.

### **Colour Ordering and the Rainbow Scale**

The following example very clearly illustrates several of the points that have been made. It shows what is meant by a visual reading or an easy interpretation at the same time as it illustrates how colour is not ordered and that value dominates colour readings. You can find these maps more extensively discussed on page 86 and 87 in Bertin's book [67/83].

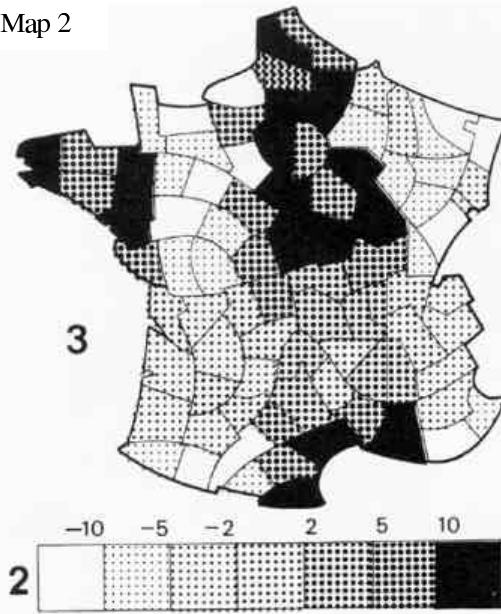
Consider the information given in numbers in map 1. One can see that there are two north-south stripes of negative numbers, one at the eastern border and one along the Atlantic excluding

# Rainbow Scale Considerations

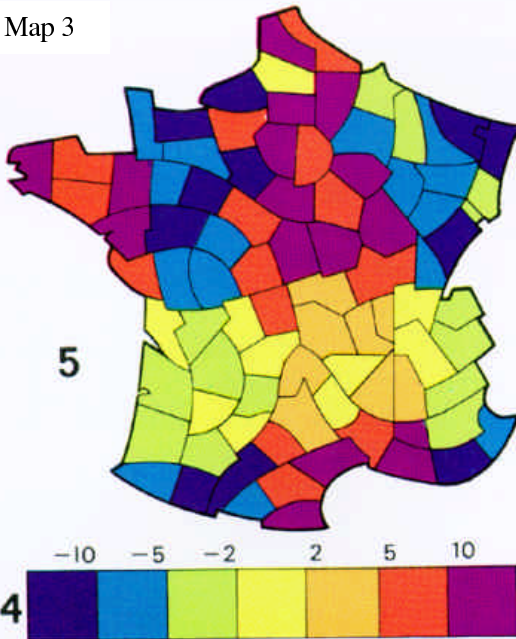
Map 1



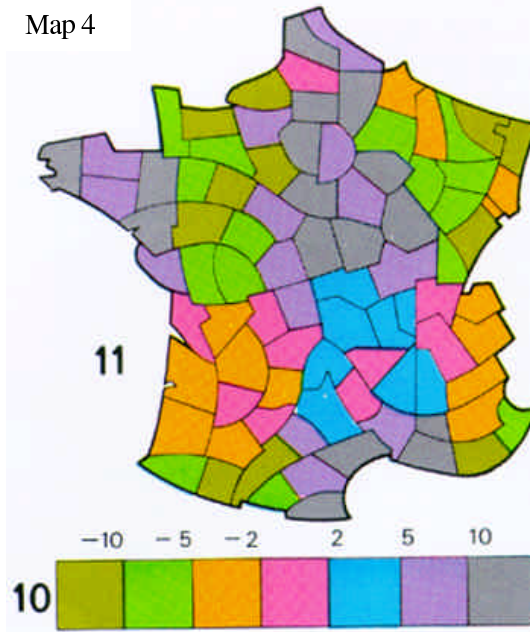
Map 2

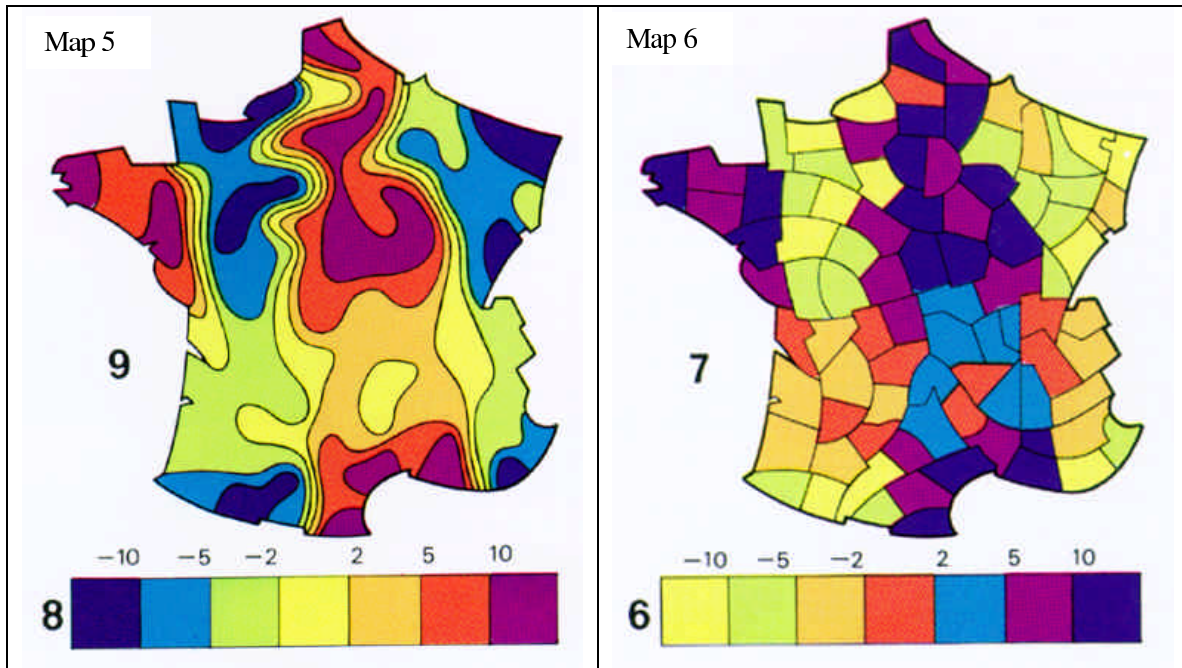


Map 3



Map 4





**Table 7: A series of maps that illustrates difficulties that arise with use of the rainbow scale**

Brittany. Map 2 represents these numbers as changes in value and the result gives a quick north-south reading, providing an easy interpretation of the information in the numbers.

Map 3 uses colour and the rainbow scale. However, this use of colour obscures the data. In fact, in map 3 it appears that the primary patterns in the data have an east-west reading. This is actually incorrect. The immediate visual interpretation follows changes in value instead of the changes in hue. The colours at each end of the rainbow scale (indigo and violet) are dark colours and the eye naturally groups them. Map 4 shows an attempt to retain the use of colour and retrieve an accurate data interpretation. Here the colours have been adjusted so that they are all of the same value (this may not be perfectly true because of colour constancy problems). This attempt at a solution provides a flat reading where distinctions between colours are not as clear. However, it does minimise the east-west reading of the full rainbow scale.

The solution provided in map 5 does use the rainbow scale and provides the correct north-south reading. However, this is done through use of more data. In this map the data values in between the high and low data values are also indicated. Access to increase resolution in the data is not always a possibility. Map 6 provides a relatively simple but effective solution by reordering the colour so their changes in value are aligned with changes in the data.

## 6.6 Orientation

The orientation of a mark that represents a point can be changed in that it be drawn in an infinite number of different orientations. Changing the orientation of an area or line can be achieved by changing in angle in which a pattern is applied (Table , row 6). These changes in alignment are discernible in 2D presentations such as printed graphics and 2D computational displays. The first limitation is that even in printed 2D graphics the shape of this mark must be distinct in terms of sections of its contour or must be longer in one direction than it is wide. This limitation becomes more acute in a computational display where orientation of lines may already be used to depict perspective. Because of this if one is using 3D computational display then using orientation is problematic.



Visual Variable: Orientation		
✓	Selective	
✓	associative	
≠	Quantitative	
≠	Order	
✓	Length	

Table 8: Visual variable orientation and interpretation tasks

**Orientation: Selective.** Orientation can be selective if the display does not use perspective and if the shape or pattern whose orientation is to be changed has a linear aspect. The answer for this visual variable is that under favourable circumstances orientation can be selective.

**Orientation: Associative.** As with selection, orientation can be associative if the display does not use perspective and if the shape or pattern whose orientation is to be changed has a linear aspect. The answer for this visual variable is that under favourable circumstances orientation can be associative. While these limitations generally apply, marks and objects that are either vertical or horizontal are quite readily seen as groups.

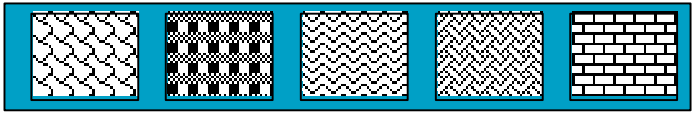
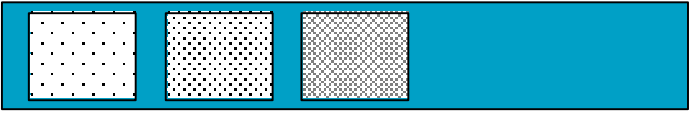

**Orientation: Quantitative.** Numerical values, quantities or ratios are not associated with changes in orientation. Changes in angles are used to represent numbers in pie charts. While we do interpret large differences in angle, it has been shown that these numerical interpretations are more difficult when represented as angles instead of length [Ware 2000].

**Orientation: Order.** There seems to be some notion of order if the changes in orientation are progressive. That is if the changes in orientation are either decreasing or increasing some sense of ordering is provided. On the other hand if they are organised randomly then this sense of order seems to dissipate. Generally it is better to assume that orientation will not be interpreted as ordered.

**Orientation: Length.** While variations in orientation theoretically infinite, practically it may be wise to limit its use to four variations, vertical, horizontal and two opposing diagonals.

## 6.7 Discussing Grain, Pattern and Texture

Bertin defines texture to be “texture variation is the sensation resulting from a series of photographic reductions of a pattern of marks” [page 79, 67/83]. This definition fits more closely with what I think of as grain. In fact [page 11, 67/83] contains a translator’s note saying that in French, Bertin uses the word grain. A change in grain is achieved by increasing the number of marks without changing the value. Defined in this manner a texture variation is a variation in granularity without a change in pattern or value or hue. For clarity in the rest of this discussion I will use the word grain where Bertin’s translator used texture. I will use the word pattern to mean repetitive use of shape (the use of marks upon marks) and keep the word texture for the apparent surface quality of the material like wood or marble. Therefore, when considering how pattern affects the visual interpretative tasks, one can turn to the discussion under the visual variable shape. The discussion that follows on grain parallels Bertin’s discussion on texture. After that I will look briefly at texture as a surface quality.

Pattern	repetitive use of shape variations	
Grain	varying granularity	
texture	a characteristic of the material	

**Table 9: Variations in pattern, grain and texture.**

## 6.8 Grain

Any colour or value except the extremes, black and white, can support variations in grain.

**Grain: Selective.** A grain can be selective; in fact, changes in grain can be very noticeable sometimes to the point of being visually irritating. Bertin suggests the careful use of this vibratory effect can provide powerful emphasis, stressing that it is the designers responsibility to ensure that this effect is not over done. [Tufte87] has several examples that over use this effect and definitely interfere with general readability.

**Grain: Associative.** Grain is minimally associative but as with selectivity the irritating effect can counteract the usefulness of this. For instance, in Table 10, row 2 the crosshatched bars can be grouped.

**Grain: Quantitative.** Grain is does not support numerical interpretations.

**Grain: Order.** Grain is not ordered expect if the changes in grain are often accompanied by changes in value.

**Grain: Length.** The larger the mark the more length there is for grain. Its use is dependent the size of the point and on the thickness of the line. Across approximately five changes in grain the interpretation tasks of selection and association are still effective.

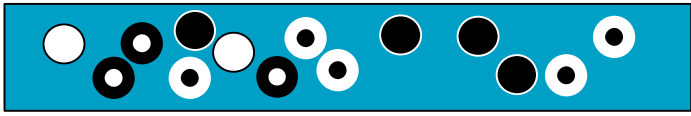
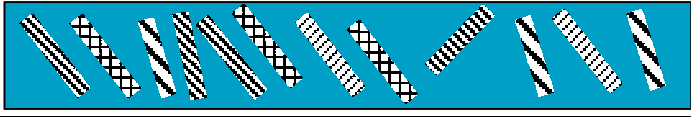


Visual Variable: Grain		
✓	Selective	
✓	associative	
≠	quantitative	
≠	order	
✓	Length	<ul style="list-style-type: none"> <li>theoretically infinite but practically limited association and selection ~ &lt; 5</li> </ul>

Table 10: Visual variable grain and interpretation tasks. This is discussed as texture in Bertin's book

## 6.9 Pattern

When considering how pattern affects the visual interpretative tasks, one can turn to the discussion under the visual variable shape.

## 6.10 Texture

Changing a mark's texture is achieved by any change in the apparent surface quality of the mark (Table 11). Note the careful distinction between pattern, which is the use of marks upon marks, grain as just discussed and the use of texture. Texture can be considered to be a surface property of the material of the mark.

**Texture: Selective.** Texture is selective since a mark changed in texture alone makes it possible to select the changed mark from all the other marks. Table 11, row 1, a change in texture from sand to grass to bricks is readily distinguishable.

**Texture: Associative.** Texture is associative since marks that are like in other ways such as shape can be grouped if they have the same texture. With texture this grouping can go further. Objects can be grouped by texture type. For instance, in Table 11 row 2, all floral textures can be thought of as a group.

**Texture: Quantitative.** There is no numerical interpretation of texture.

**Texture: Order.** Are changes in this texture are not perceived as ordered unless they are associated with changes in value.

**Texture: Length.** The visual variable texture has considerable length. There are not only many different textures there are many types of different textures.




Visual Variable: Texture		
✓	selective	
✓	associative	
≠	quantitative	
≠	order	
✓	Length	<ul style="list-style-type: none"> <li>theoretically infinite</li> </ul>

Table 11: Visual variable texture and interpretation tasks

### 6.11 Motion

In the physical world we are very sensitive to motion and can detect changes in motion even in our peripheral field of vision. In printed graphics use of motion was not a possibility. In computational displays, while it is a possibility, it has been little used in information representation and little investigated. There are many aspects to motion that can be changed. These include: direction, speed, flicker, frequency, rhythm, style, onset, etc. The following is intended to be just preliminary discussion.

**Motion: Selective.** Since motion is one of our most powerful attention grabbers it is probably selective.

**Motion: Associative.** Objects moving in unison are grouped effectively. In fact, objects moving in unison may be thought of as being united.

**Motion: Quantitative.** It is likely that there is no numerical reading obtainable from changes in this motion. However, changes in speed or frequency or other characteristics of motion may prove to be distinguishable in this manner.

**Motion: Order.** Motion may be ordered. Speed may be one characteristic by which motion can be ordered.

**Motion: Length.** There are a considerable variety of motions.

### Discussion of the Use of Visual Variables on a Computer

This report has discussed Bertin's concepts about visual variables [Bertin 67/83] in context of their use in information visualisation on computational displays. Though the distinctions and variations use of computer brings to this concept has been mentioned; it has not yet been fully investigated. Some of these differences include:

- The addition of surfaces and volumes to types of marks,
- The addition of the third dimension in positional variables,
- The separation of colour into hue, saturation and value. (in this document changes in hue are discussed as changes in colour, changes in value are discussed as changes in value and changes in saturation are not discussed.)

- The visual variable orientation loses much of its usefulness because of its interference with perspective presentations.
- I have left pattern, as did Bertin [Bertin 67/83], to be considered as repetitive variation in shape. This should be re-assessed.
- Since use of texture is so readily available in computational displays, I have reverted to the term Bertin [Bertin 67/83] used in French ‘grain’ and introduced texture as a change in surface quality.
- I have just touched on the possibilities of motion.
- There are other possibilities such as depth, occlusion, and transparency, which should be addressed.

## References

- [Bartram97] Lyn Bartram. Perceptual and interpretative properties of motion for information visualization. *Technical Report CMPT-TR-1997-15, School of Computing Science, Simon Fraser University, Burnaby, BC, Canada, October 1997.*
- [Bertin67/83] Jacques Bertin. *Semiology of graphics: diagrams, networks, maps*. University of Wisconsin Press, 1983. (first published in french in 1967 translated by William J. Berg in 1983).
- [Dondis73] Donis A. Dondis. *A primer of visual literacy*. MIT Press, Cambridge, Mass., 1973.
- [Fiske90] John Fiske. *Introduction to communication studies. Studies in culture and communication*. Routledge, London ; New York, 2nd edition, 1990.
- [Horn98] Robert E. Horn. *Visual Language*. MacroVu Inc. Washington, 1998.
- [Marr82] D. Marr. *Vision: A computational investigation into the human representation and processing of visual information*. W. H. Freeman and Company, 1982.
- [McCloud93] Scott McCloud. *Understanding Comics: The invisible art*. Harper Collins, New York, New York, 1993.
- [Tufte83] Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, Connecticut, USA, 1983.
- [Tufte90] Edward R. Tufte. *Envisioning Information*. Graphics Press, Cheshire, Connecticut, USA, 1990.
- [Tufte97] Edward R. Tufte. *Visual Explanations: Images and Quantities, Evidence and Narrative*. Graphics Press, Cheshire, Connecticut, USA, 1997.
- [Ware93] C. Ware. The foundations of experimental semiotics: a theory of sensory and conventional representation. *Journal of Visual Languages and Computing*, 4:91–100, 1993.
- [Ware94] C. Ware and G. Franck. Viewing a graph in a virtual reality display is three times as good as a 2Ddiagram. *In IEEE Conference on Visual Languages*, pages 182–183, October 1994.
- [Ware00] Colin Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann, 2000.

# Externalising Abstract Mathematical Models

Lisa Tweedie, Robert Spence, Huw Dawkes and Hua Su

Department of Electrical Engineering,  
Imperial College of Science, Technology and Medicine  
South Kensington, London, SW7 2BT  
Tel: +44 171 594 6261  
l.tweedie@ic.ac.uk

## ABSTRACT

Abstract mathematical models play an important part in engineering design, economic decision making and other activities. Such models can be externalised in the form of Interactive Visualisation Artifacts (IVAs). These IVAs display the data generated by mathematical models in simple graphs which are interactively linked. Visual examination of these graphs enables users to acquire insight into the complex relations embodied in the model. In the engineering context this insight can be exploited to aid design. The paper describes two IVAs for engineering design: The Influence Explorer and The Prosection Matrix. Formative evaluation studies are briefly discussed.

**KEYWORDS:** Interactive Graphics, Visualization

## INTRODUCTION

Many mathematical problems can benefit from being examined visually. Indeed most spreadsheets and statistical packages enable users to quickly create static representations of their data. These graphs have an accepted role as tools for mathematical problem solving. However the value of adding interactivity to such representations has yet to gain widespread recognition.

Responsive (i.e. rapid) interaction can facilitate active exploration of problems in a manner that is inconceivable with static displays. For example users can start to pose "What if" queries spontaneously as they work through a task. Such exploration can enormously facilitate the acquisition of qualitative insight into the nature of the task at hand, as well as revealing direct quantitative results.

In this paper we describe what we call Interactive Visualisation Artifacts (IVAs). These are environments developed to enable users to solve a particular task - in this case within the field of engineering design.

The IVAs we will discuss here differ from much existing work principally because we are not attempting to visualise

Permission to make digital/hard copies of all or part of this material for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copyright is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires specific permission and/or fee.

CHI 96 Vancouver, BC Canada

© 1996 ACM 0-89791-777-4/96/04..\$3.50

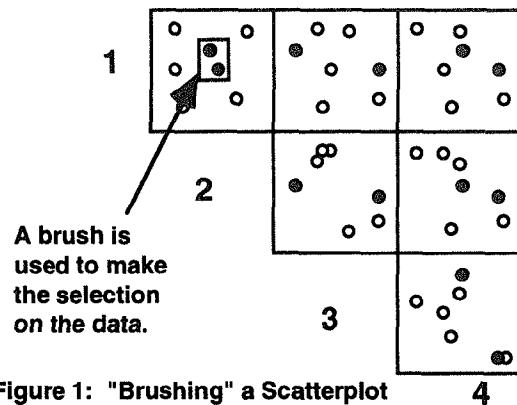


Figure 1: "Brushing" a Scatterplot

raw data but, rather, data which is precalculated or generated on demand from mathematical models. We also exclude data which maps comfortably onto natural representations e.g. 3D volumetric models of flow through a pipe. Instead we focus on more *abstract* mathematical models which have no obvious representation.

We can take as an example the design of an engineering artifact. Mathematical models (equations) exist which relate the artifact's performance to the parameters that describe the physical nature of that artifact. Thus, for a bridge, performances such as traffic capacity and cost can be calculated from a knowledge of parameters such as cable diameters and foundation depth. A designer needs to explore the relationships between parameters and performances in order to elicit a useful design.

The development of IVAs for such applications requires the creation of new representations that externalise pertinent aspects of the model. The IVAs we describe in this paper show how such novel representations can be created by **interactively linking simple graphs in several ways**. On a simple level we can link many similar graphs, as Becker et al [3] did with their "brushed" scatterplots (Figure 1). We can also link different *types* of representations together. For example, by selecting a subset of data on a histogram and colour encoding the same subset on a scatterplot. These links can also perform different functions - for example the selected subset could be colour encoded or it could be hidden from view.

Two IVAs for engineering design are described in this paper: the Influence Explorer and the Prosection Matrix. They exhibit powerful and effective linking both within and between IVAs.

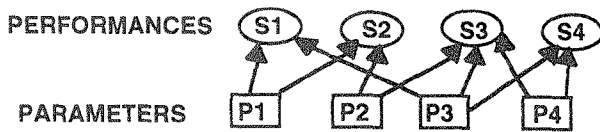


Figure 2: The Parameter->Performance relationship

**Previous Work**

The idea of linking graphical representations is not new. As early as 1978 Newton [12] was linking several scatterplots and colour encoding selections to discover trends in data. Many others have developed simple linking IVAs e.g. IVEE [1], Permutation Matrices [4], BEAD [5], SeeSoft™[6], AutoVisual [7], VisDB [10], Nested Histograms [13], The Table Lens [14], Visulab [15], The InfoCrystal [17], The Attribute Explorer [18] and The Dynamic HouseFinder [20].

Most of these IVAs only use one type of representation to display data. However a combination of representations may also be beneficial, since the user is then able to consider the problem from several different perspectives. Schmid and Hinterberger [15] have called this "Comparative Multivariate Visualisation" and embodied the concept in their "Visulab" software. Here four different representations (Parallel Coordinates [9], Andrews Plots [2], Permutation Matrices [4] and Multiple Scatterplots [3]) can be linked in several ways : encoding with colour, hiding part of the data and reordering the data. The use of several different representations of data, and the manner of their linking, is a key issue in the development of IVAs.

**Visual Design Issues**

The design of any IVA should proceed with various characteristics of visual problem solving in mind (Tweedie [19]). As Nardi and Zamer [11] point out, IVAs are external representations of the users problem which "stimulate and initiate cognitive activity". Zhang and Norman [21] identify that such external representations act as memory aids; provide information perceptually without need for interpretation; anchor and structure cognitive behaviour; and change the task.

Suchman [18] emphasises that "it is frequently only on acting in a present situation that its possibilities become clear". In other words users will often pick up information opportunistically from their environment. It is partly this tendency to stimulate opportunistic behaviour that makes IVAs interesting. Consequently, the visual cues provided must be designed to support this opportunistic process.

**DESIGN FOR MANUFACTURABILITY**

A typical task that has a mathematical model associated with it is that of engineering design. For a given product such as a light bulb, a model can be formed of the way the parameters (whose value is open to choice by the designer) influence performances (Figure 2). In the light bulb example, performances such as a bulb's brightness and its lifetime will partly be determined by parameters such as the number of coils in its filament and the thickness of that filament. The mathematical model is a set of equations, each relating a performance to a number of parameters.

The designer must choose numerical values for parameters in such a way that the performances they influence, usually in a very complex fashion, take on values acceptable to a customer. In other words, when designing a light bulb, the designer has to keep a specification in mind. If for example they are asked to design a light bulb that will be very bright and last for at least 6 months then they need to find the set of parameters values that will satisfy this specification.

**The traditional design process**

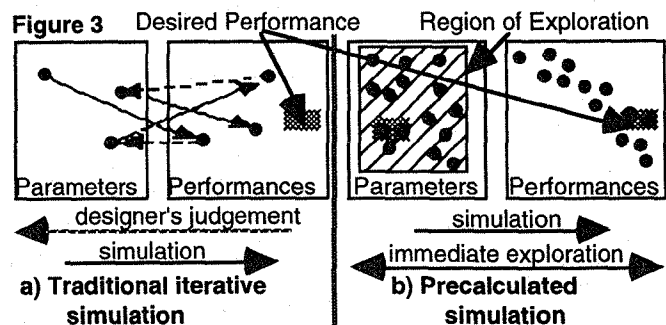
Given a set of parameters, an engineering artifact can be simulated to establish the corresponding performances. Unfortunately the reverse is not true: a designer cannot choose a performance value and calculate the parameters needed to achieve it. For this reason traditional design is characterised by a series of iterations in which the designer selects a set of parameters and then simulates the artifact to find out what the performances are. Design proceeds through the gradual adjustment of parameters until a satisfactory set of performance values is found. This design process is illustrated in figure 3a for an artifact defined by two parameters and influencing two performances. The design is represented by a single point moving in parameter and performance space. This "trial and error" approach can be tedious and time-consuming and is heavily dependent on a designer's expertise.

**Precalculation**

The design process can be immensely simplified if one has mathematical models of the relationship between parameters and performances. Figure 3b shows how such models can be used to create a precalculated exploration database. The designer selects a wide "Region of Exploration" in parameter space within which the final design might well be expected to lie. Within this region a large number of points (e.g. over 500) are generated randomly, each point representing a design. For each of these sets of parameter values the corresponding point in performance space is computed using the artifact's mathematical model. In our light bulb example, a dataset generated in this way would describe a variety of light bulbs each having randomly different parameter values and associated performances. The benefit of creating such a dataset is also illustrated in figure 3b. The designer can now readily select their desired performance values and "look up" which parameter sets give them those values.

**Designing in the real world**

Unfortunately the aim of engineering design is not simply that of finding a single set of parameter values that satisfies



a specification. Inevitable fluctuations in manufacturing processes mean that parameter values can only be guaranteed to lie within a so-called *tolerance range*. For example the filament width of our light bulb might vary slightly during manufacture, and this variation could have a crucial effect on a performance. We therefore need to define exactly how much each parameter can vary. The combined set of parameter tolerance ranges defines a *tolerance region* in parameter space. These are the bulbs that will be manufactured.

Figure 4 shows the rectangular tolerance region for the simple case of two parameters. In the same space, an irregularly shaped "Region of Acceptability" defines the location of all the artifacts that satisfy the performance requirements. Achieving a good design is a matter of fitting these two regions to each other with maximum overlap

**Overall Design Objectives**

As well as satisfying the customer's requirements on performance, it is usually the case that there is also some overall objective that must be achieved. One such objective is that of maximising the manufacturing yield, which is the percentage of mass-produced bulbs that satisfy the customer's requirements on performance. With reference to Figure 4, yield is that percentage of the tolerance region which lies within the region of acceptability.

Another such design objective might be the unit manufacturing cost of each bulb that is shipped to the customer. Usually the wider the tolerances are on the parameters the cheaper the bulb will be to manufacture.

**THE INFLUENCE EXPLORER**

Precalculation forms the backbone of the Influence Explorer. Once the data has been precalculated (as described earlier), it provides an exploration database on which to start an investigation. Figure 5 shows how the population of 600 precalculated designs is displayed in the form of histograms. All performance histograms are plotted horizontally to the left of the screen and the parameter histograms vertically to the right. An artifact is represented once on each plot in the appropriate bin. Each column in the histogram represents the number of designs that fall within that bin. In other words, the histograms are frequency plots.

**Qualitative Exploration**

In order to form an effective external representation of the task the Influence Explorer must allow the user to gradually

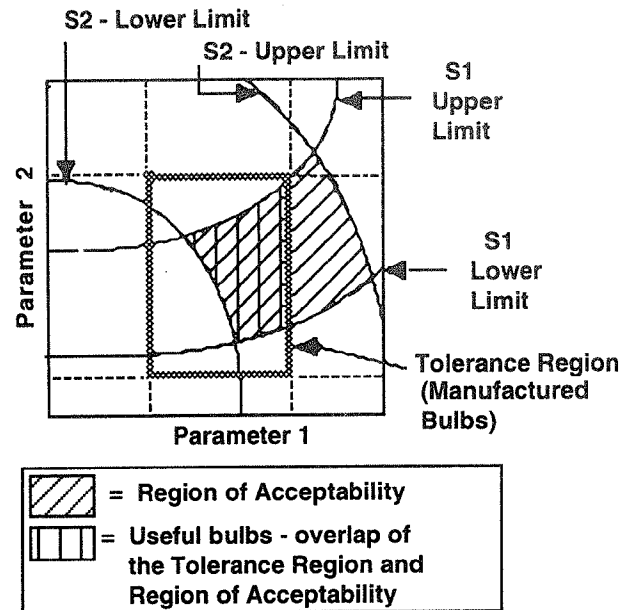


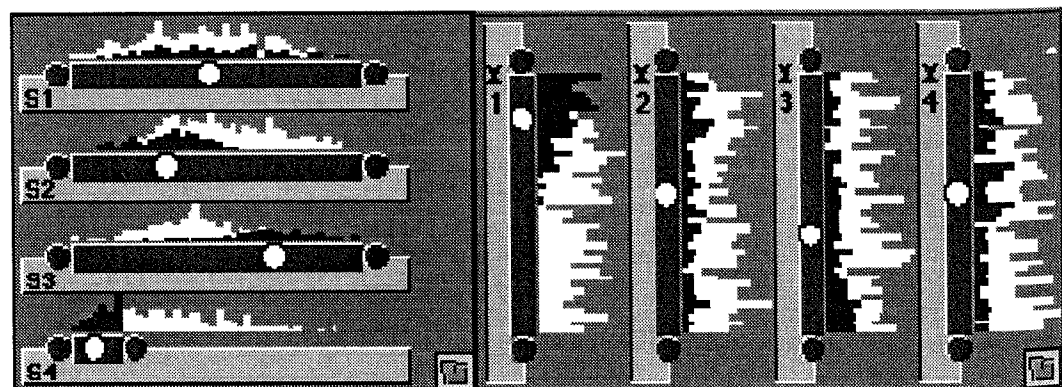
Figure 4: In 2-parameter space, two performances F1 and F2 define upper and lower limits for acceptable performance. Manufactured bulbs lie within the tolerance region.

build up a coherent picture of their problem, in other words the complexity must be introduced in stages.

In the initial stages of design the user will want to gain a *qualitative* understanding of the problem. The designer can place exploratory limits on parameters and performances, thereby defining ranges of those quantities. In Figure 5 a range of performance on S4 has been defined with a slider. This action leads to the colour linking (black) of those bulbs that lie within the selected range on the S4 histogram and all the other histograms, so that the selected subset can be viewed across all the histograms. The potential for exploring the inter-relation between parameters and performances is now apparent. Confidence in these perceived relations can be sought by interactively moving the selected range of S4 up and down its scale and observing the corresponding movement of the highlighted bulbs on the other scales. The power of such a dynamic action to generate insight is difficult to convey in static words and diagrams, but is *strikingly* obvious in actual use.

It is worth emphasising that the discovery of a "trade-off" relation between two performances is immensely important

Figure 5: The performance (left) and parameter (right) histograms. A selection has been made on S4 and these same points are highlighted on each of the other histograms. Circles indicate the mean of the selected points.





in engineering design. In the Influence Explorer this discovery is virtually immediate, whereas in conventional design practice such a trade-off might be discovered only after tedious search or, at worst, not at all.

Additional tools enhance the functionality of the Influence Explorer. A mouse-click on a bulb in one histogram highlights that same bulb, and displays corresponding values, in all the other histograms. Another option connects these points with a line and allows the comparison of several different bulbs. These lines are known as "parallel coordinate" plots [9]. Yet another option places a circle on each of the histogram scales indicating the mean of the currently selected bulbs (see figure 5). This is useful when a range is being moved as it eases detection of trends.

### Quantitative Design decisions

As well as indulging in *qualitative* exploration, the designer must at some stage take note of the *quantitative* detail associated with a customer's requirements on performance. To do so a "specification option" is selected (Figure 6 - colour plate).

The placement of upper and lower limits on the performance scales invokes another linking mechanism. Red colour coding identifies bulbs that lie within all the performance limits, those bulbs which fail one limit are colour coded black, while dark and light grey denotes two and three failed limits respectively. Such colour coding provides valuable sensitivity information. For example, it is immediately noticed (Figure 6 - colour plate) that a relaxation of the upper limit on S4 would turn some black bulbs into (acceptable) red bulbs, knowledge which might well lead to a discussion about the wisdom of that particular upper limit. Negotiations concerning performance specifications are common to engineering and could be considerably clarified using this information.

### Design for Manufacture

As already explained, inevitable variations in the manufacturing process are such that, in the design of a *mass-produced* artifact such as a light bulb, the designer must be concerned with the selection of parameter *ranges* rather than specific values. It is the combination of all these selected parameter ranges that must satisfy the performance limits defined by the customer.

Parameter ranges are defined by the selection of upper and lower limits (Figure 7 - colour plate), in exactly the same manner as for the performances. Again, the selection of parameter limits invokes a linking mechanism, once more leading to additional colour encoding. Though at first sight complex, the coding is, we suggest, matched to an engineering designer's real needs and, given the motivation provided by a tool offering responsive exploration, is readily, even eagerly learned. Figure 8 (colour plate) is a replica of Figure 4 with the relevant colour codings shown. Figure 8 and the table attached to figure 7 may help clarify the rationale behind this coding:

- **Red** denotes bulbs that satisfy all limits. They lie within parameter limits (and are therefore manufactured) and they satisfy the customer's performance limits.

- **Black** denotes a bulb that satisfies all the performance limits but lies outside one parameter limit, and is therefore not manufactured. Thus it will turn red if one parameter limit is adjusted to include it.

- **Blue** bulbs are those which are manufactured (and hence lie within parameter limits) but fail one or more performances. These are the bulbs which cause a reduction in yield. Tightening a parameter limit to eliminate blue bulbs (for example raising the lower limit of X1 in Figure 7) will reduce the number of manufactured artifacts which violate a customer's requirements, hence raising the yield. The Blue bulbs are coded in two shades of blue - **Dark Blue** indicates those bulbs that are manufactured and only violate one performance limit; relaxation of that performance limit will turn those bulbs into red ones (e.g. in figure 7 expanding the lower limit on S1 will turn the dark blue bulbs red). **Light Blue** indicates those bulbs which are manufactured and violate more than one performance limit.

- **Grey** bulbs are those which fail one parameter range and one or more performance limits. They would therefore turn blue if they were to be enclosed within the tolerance region. Thus in Figure 7 if the upper limit on X2 is extended to turn the black bulbs into red ones, this gain in the number of (red) acceptable bulbs would be offset by the number of grey bulbs turning blue and, thereby, adding unsatisfactory bulbs to the manufacturing process.

The principal advantage of such colour coding is that it indicates how *altering* the parameter or performance limits will effect the overall usefulness of the design.

### Yield Enhancement

To facilitate design for maximum yield the Influence Explorer continuously computes, and displays in numerical form, the value of the yield. The designer may well begin by attempting to select parameter ranges that maximise the yield, hopefully to a value of 100%. In order to achieve such a high yield the user needs to adjust the tolerances taking account of where the red and therefore "useful" points lie and trying to reduce the number of blue points. By keeping an eye on the yield the user can slowly optimise their solution until they have found an optimum yield.

100% yield can obviously be achieved by making the parameter ranges sufficiently small (Figure 14 - colour plate), but another overall objective - the minimisation of manufacturing cost - militates against such a solution. It is normally the case that the wider the parameter ranges, the lower the cost of the artifact. There is therefore a strong incentive to select parameter ranges that are as wide as possible commensurate with an acceptably high yield (see Figure 15 - colour plate).

### Focused Sampling

Unfortunately when interacting with tolerances limits the precalculated data set becomes a constraining factor in the Influence Explorer. Since the requirements are now becoming specific, it is unlikely that many of the original 600 points will fall within *all* the performance and parameter requirements. This curse of dimensionality results in very few colour coded points. To overcome this problem the Influence Explorer is programmed to

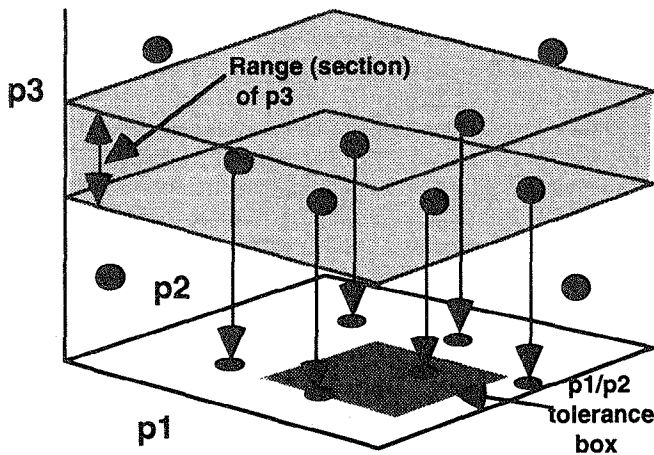


Figure 9: A section of p3 is projected onto a p1/p2 scatterplot

dynamically resample the model so that a number of points always fall within and close to the tolerance region. Evidence of this process can be seen in Figure 7 where the column heights within the tolerance limits are higher than column heights on the rest of each parameter histogram.

**THE PROJECTION MATRIX**

The Projection Matrix provides an alternative perspective of the model. It is a set of scatterplots (Figure 10) arranged in a matrix, as suggested by Becker et al [3]. Each scatterplot corresponds to a different pair of parameters, and all possible parameter pairs are represented. Thus, for the bulb's four parameters there are six scatterplots.

The construction of each scatterplot is illustrated conceptually in Figure 9 for the simple case of a 3-parameter system. p1 and p2 are the scatterplot's two parameters. p3 is a third parameter on which a parameter range has been set. Only data that falls within p3's chosen parameter range is projected down onto the p1p2 plane.

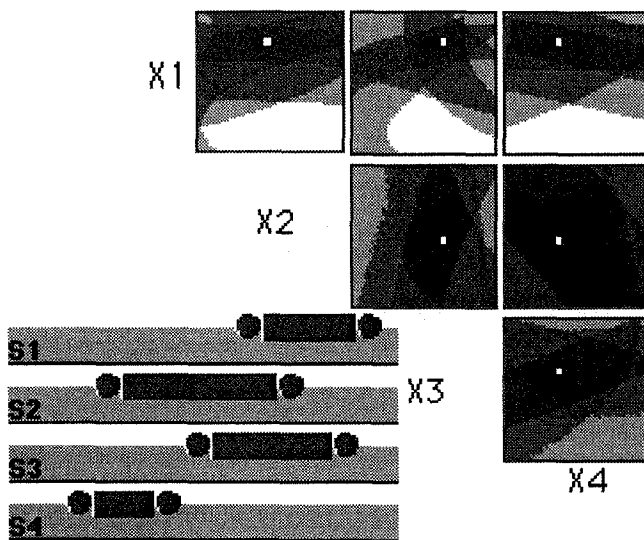


Figure 10: This Projection Matrix represents 'slices' through parameter space. The grey scaling show how the data satisfies the performance requirements.

This is a projection of a section of parameter space, hence the name 'Projection' (the term came from by a paper by Furnas and Buja [8]). This projection process is repeated for every pair of parameters so that each scatterplot is displaying different data. The tolerance ranges for the scatterplots two parameters (p1 and p2 in figure 9) can also be projected on to the plot in the form of a tolerance box.

The Projection Matrix shown in Figure 10 actually refers to a situation in which each parameter range is very small, leading to a small tolerance region (the small grey dot in the centre of each scatterplot). Because the parameter ranges are small, they define a very thin 'slice' through multi-dimensional parameter space, and therefore the resulting scatterplots show well-defined boundaries associated with the different performance limits of Figure 10. The colour coding used defines how well designs satisfy these performance limits. In Figure 10 designs that are acceptable are black, those that failed only one performance limit are dark grey and those that fail two are medium grey etc. One of the benefits of this colour coding is that the designer can explore the effect of moving the boundaries in the scatterplot. Thus, in Figure 11, the designer has moved the lower limit of performance S3 even lower. A comparison of Figure 10 and 11 reveals how the corresponding boundary has moved, increasing the area of the (here, black) acceptable region. Exploration of this kind allows a designer to form a strategy for combining and trading off different performance requirements.

Though Figure 9 provides a conceptual illustration of the formation of each scatterplot within the Projection Matrix it is actually unsuitable for implementation because it would result in a very grainy representation. Instead, each scatterplot is filled using a matrix of small coloured squares. For example if we consider the (top left) X1X2 scatterplot in Figure 11 its area is divided in 44<sup>2</sup> squares, The Cartesian coordinates of each square's midpoint defines values of X1 and X2. Values of X3 and X4 are then

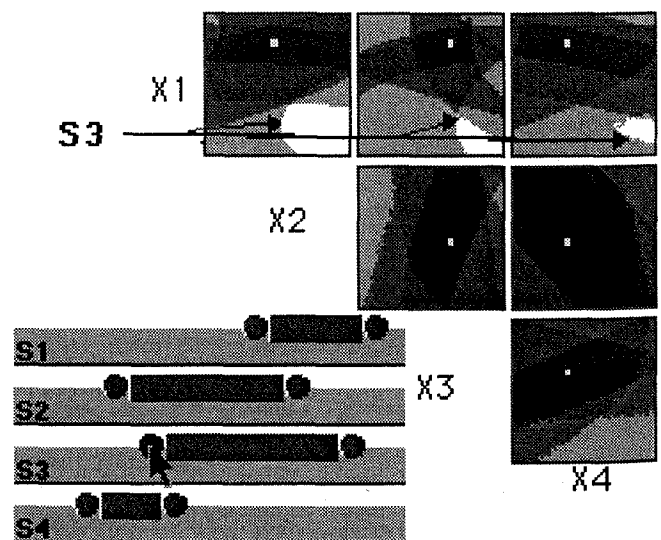


Figure 11: Adjusting a performance requirement and viewing how the related boundary moves in parameter space

selected randomly from within their tolerance range for each square. The corresponding values of the performances S1 to S4 are then computed from the model and compared with their respective limits. The square is then coloured according to the scheme already defined. For clarity, in the case of Figure 11 the X3 and X4 ranges are actually set to a single point so no randomisation occurs.

The existence of significant parameter *ranges* rather than single parameter values changes the detailed appearance of the Projection Matrix but not its general character (Figure 12). Again consider the X1X2 scatterplot (top left). The original value of X1 has been replaced by a range of X1 as indicated by the yellow line. The immediate effect is that for all the scatterplots that don't have X1 as an axis, X1 is now randomly chosen within the selected *range* of X1 values rather than set at a single value. The increased fuzziness of these plots reflects this process. The rest of Figure 12 shows the effect of additionally assigning ranges to X2, X3 and X4.

Figure 13 (colour plate) shows how the Projection Matrix looks when the performance and parameter limits are set as in Figure 7. The red regions now correspond to acceptable bulbs, whereas those that are manufactured lie within the yellow tolerance regions. The small percentage of red points within this region indicate a low yield (19%). In Figure 14 (colour plate) the user has set the tolerances to very narrow ranges to find a high yield (100%). Since wider

tolerance ranges are normally associated with lower cost, the designer will endeavour to make the yellow -bounded tolerance region as large as possible, perhaps even trading off manufacturing yield against cost. Figure 15 (colour plate) shows how the user has adjusted the parameter ranges so that they just fit inside the red region, resulting in much wider tolerances (potentially cheaper components) whilst maintaining a reasonably high yield (96%)

**FORMATIVE EVALUATION STUDIES**

The design of IVAs is difficult - it is often hard to judge what users will find intuitive and how an IVA will support a particular task. We have therefore carried out a number of formative evaluation studies at different stages of the IVA's development. Ten pairs of subjects were tested. They were all graduate engineers/ scientists enrolled on PhD programs. The pairs worked together, first with the Influence Explorer, then the Projection Matrix and finally both tools together. Reassuringly, each pair of subjects were able to complete a tolerance design task in about 30 minutes.

We learnt some very simple lessons from these evaluations:  
 a) *Maximise the directness of the interactivity.* For example one version of the Projection Matrix forced users to map their interaction from the sliders. However users preferred to select and drag the tolerance box directly.  
 b) *Seek out the most crucial information and then represent it appropriately and simply.* The most obvious example of

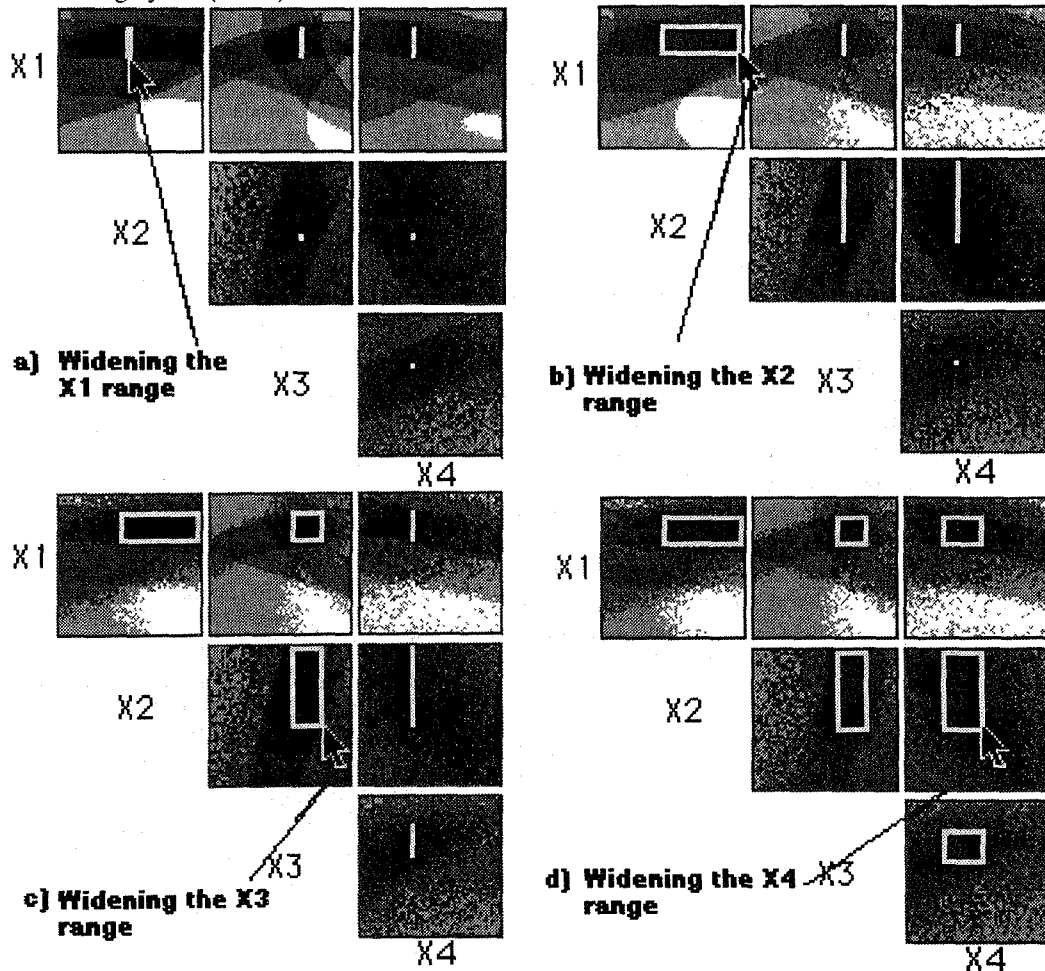


Figure 12: Gradually increasing the tolerance region so that sections of the data are projected. The boundaries become fuzzier as the ranges are adjusted.

this was the colour coding. Initially when considering the interface for setting up a performance specifications we attempted to colour code all the different variations of failure. Then we realised that this coding could be considerably simplified if we focused on encoding data that satisfied the performance limits and perhaps more importantly data that *almost* satisfied those limits. Colour coding the influence explorer for tolerance design was more difficult. The solution presented in this paper (Figure 8) has attempted to reduce the colour coded information to that which will provide immediate and useful information.

c) *There is a trade-off between the amount of information, simplicity and accuracy.* Ensuring that there is sufficient information to complete a task was an important issue. This emerged in the Influence Explorer when we tried adding tolerances with the original precalculated dataset. Using dynamic focused sampling overcame this problem.

## CONCLUSIONS

The Influence Explorer and Prosection Matrix have now been utilised in a wide variety of industrial collaborations in electronic, structural and mechatronic domains. The enthusiastic reaction of those who have observed and experimented with these IVAs suggests that the potential offered by immediately available and responsive interaction is considerable.

There are many reasons for this enthusiasm. One is the readiness with which opportunistic as well as planned exploration can be carried out. Another is the directness of external representations. Abstract Mathematical Models are difficult for the untrained user to interpret. However using these IVAs the problem holder can explore the model for themselves, and make use of their own considerable experience and knowledge to test the models validity in their own terms. A mathematical model is one thing, but an externalisation of that mathematical model that can be responsively explored is quite another. A third reason is that these tools transform a very difficult cognitive problem into a much easier perceptual task.

Many avenues of research and experimentation still need to be followed up. One concerns the enhancement of the designer's expertise by some of the automated tolerance design algorithms developed over the last two decades. One such algorithm was incorporated within the Influence Explorer and, when invoked, automatically and very rapidly (e.g. 10 seconds) adjusted the 'nominal value' of each parameter (the mid-point of the selected parameter range) to maximise the yield. Nevertheless, this automation needs to be complemented by an interface which will facilitate the human observation and guidance of automated design.

## ACKNOWLEDGEMENTS

We thank Aarnout Brombacher (Philips, Eindhoven) and John Nelder. This work was implemented in C on a Macintosh by Huw Dawkes.

## REFERENCES

[1] Ahlberg C. and Wistrand E., "IVEE: An Environment for Automatic Creation of Dynamic Queries Applications," CHI'95 Demonstrations, May 1995.

[2] Andrews D.F. "Plots of High-Dimensional Data" Biometrics, March 1972, pp 69-97

[3] Becker R.A., Huber P.J., Cleveland W.S. and Wilks A.R., "Dynamic Graphics for Data Analysis", Stat. Science 2, 1987.

[4] Bertin J., "Graphics and Graphic Information Processing", deGruyter Press, Berlin, 1977.

[5] Chalmers M., "Using a Landscape to represent a corpus of documents", Springer-Verlag Proceedings of COSIT '93, Elba, pp. 377-390, September 1993.

[6] Eick S.G., Steffen J.L. and Sumner E.E., "SeeSoft™ - A Tool for Visualizing Line Oriented Software", IEEE Transactions on Software Engineering, pp. 11-18, 1992.

[7] Feiner S. and Beshers C. "Worlds within Worlds: Metaphors for Exploring n-Dimensional Virtual Worlds", ACM Proceedings 1990 Conference on User Interface Software Design, pp 76-83

[8] Furnas G.W. and Buja A., "Prosection Views: Dimensional Inference through Sections and Projections", Journal of Computational and Graphic Statistics 3 (4), pp. 323-353, 1994.

[9] Inselberg A., "The plane with parallel co-ordinates", The Visual Computer 1, pp. 69-91, 1985.

[10] Keim D.A. and Kriegel H., "VisDB: Database Exploration using Multidimensional Visualization", IEEE Computer Graphics and Applications September, pp. 40-49, 1994.

[11] Nardi B.A. and Zamer C.L., "Beyond Models and Metaphors: Visual Formalisms in User Interface Design", Journal of Visual Languages and Computing 4, pp. 5- 33, 1993.

[12] Newton C.M., "Graphics: from alpha to omega in data analysis", Graphical Representation of Multivariate Data, P.C.C. Wang (Ed) New York: Academic Press, pp. 59-92, 1978.

[13] Mihalisin T., Gawlinski E., Timlin J. and Schwegler J., "Visualizing Scalar Field on an N-dimensional Lattice", Proceedings of Visualization 90, IEEE CS Press, pp. 255-262, 1990.

[14] Rao R. and Card S.K., "The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus + Context Visualization for Tabular Information", Proceedings of CHI'94, Boston, ACM Press, pp. 318-322, 1994.

[15] Schmid C. and Hinterberger H., "Comparative Multivariate Visualization Across Conceptually Different Graphic Displays", Proceedings of the SSDBM VII, IEEE Computer Society Press, September 1994.

[16] Spoerri A., "InfoCrystal: A visual tool for Information retrieval" Proceedings of Visualization '93 pp150-157.

[17] Suchman L.A., "Plans and Situated Actions - The Problem of Human-Machine Communication", Cambridge University Press, 1987.

[18] Tweedie L.A., Spence R., Bhoghal R. and Williams D., "The Attribute Explorer", ACM, Video Proceedings and Conference Companion, CHI'94, pp. 435-436, April 1994.

[19] Tweedie L.A., "Interactive Visualisation Artifacts: how can abstractions inform design?", People and Computers X : Proc. of HCI'95 Huddersfield, (Eds) Kirby M.A.R., Dix A.J. and Finlay J.E., Cambridge University Press, pp. 247-265, 1995.

[20] Williamson C. and Shneiderman B., "The Dynamic HomeFinder: Evaluating dynamic queries in a real estate information exploration system", ACM, Proceedings SIGIR'92, pp. 339-346, 1992.

[21] Zhang J. and Norman D.A., "Representations in Distributed Cognitive Tasks", Cognitive Science 18, pp. 87-122, 1994.

# Toward a Deeper Understanding of the Role of Interaction in Information Visualization

Ji Soo Yi, Youn ah Kang, John T. Stasko, *Member, IEEE*, and Julie A. Jacko

**Abstract**—Even though interaction is an important part of information visualization (Infovis), it has garnered a relatively low level of attention from the Infovis community. A few frameworks and taxonomies of Infovis interaction techniques exist, but they typically focus on low-level operations and do not address the variety of benefits interaction provides. After conducting an extensive review of Infovis systems and their interactive capabilities, we propose seven general categories of interaction techniques widely used in Infovis: 1) Select, 2) Explore, 3) Reconfigure, 4) Encode, 5) Abstract/Elaborate, 6) Filter, and 7) Connect. These categories are organized around a user's intent while interacting with a system rather than the low-level interaction techniques provided by a system. The categories can act as a framework to help discuss and evaluate interaction techniques and hopefully lay an initial foundation toward a deeper understanding and a science of interaction.

**Index Terms**—Information visualization, interaction, interaction techniques, taxonomy, visual analytics

## 1 INTRODUCTION

Information visualization (Infovis) systems, at their core, appear to have two main components: representation and interaction. The representation component, whose roots lie in the field of computer graphics, concerns the mapping from data to representation and how that representation is rendered on the display. The interaction component involves the dialog between the user and the system as the user explores the data set to uncover insights. The interaction component's roots lie in the area of human-computer interaction (HCI). Although discussed as two separate components, representation and interaction clearly are not mutually exclusive. For instance, interaction with a system may activate a change in representation. Nonetheless, the two components seem to compose the two fundamental aspects of Infovis systems, and it seems reasonable to consider what each contributes to an end-user's experience.

We argue that the representation component has received the vast majority of attention in Infovis research. A cursory scan of a recent conference proceedings or journal issues in the area will uncover many articles about new representations of data sets, but interaction is often relegated to a secondary role in these articles. Interaction rarely is the main focus of research efforts in the field, essentially making it the "little brother" of Infovis. In other words, it is overshadowed by the more noteworthy representation aspects. A few papers have mainly focused on the interactive aspects of Infovis (e.g., [10, 15, 25, 47]), but these are relatively uncommon when compared to papers introducing new data representations.

Interaction is an essential part of Infovis, however. Without interaction, an Infovis technique or system becomes a static image or autonomously animated images (e.g., InfoCanvas [28]). While static images clearly have analytic and expressive value (e.g., [8, 29, 46]), their usefulness becomes more limited as the data set that they represent grows larger with more variables. Actually, even with a static image such as a poster, a user (or a reader) will often perform several interactions (e.g., rotating the poster, looking closer/further,

and jotting down notes on the poster). Spence even suggests the notion of "passive interaction" through which the user's mental model on the data set is changed or enhanced [38]. Finally, through interaction, some limits of a representation can be overcome, and the cognition of a user can be further amplified (e.g., [15, 29]).

The importance of interaction and the need for its further study seem undisputed. For example, the recent book *Illuminating the Path: The Research and Development Agenda for Visual Analytics* calls for further research on interaction:

**"Recommendation 3.3: Create a new science of interaction to support visual analytics.** The grand challenge of interaction is to develop a taxonomy to describe the design space of interaction techniques that supports the science of analytic reasoning. We must characterize this design space and identify under-explored areas that are relevant to visual analytics. Then, R&D should be focused on expanding the repertoire of interaction techniques that can fill those gaps in the design space." ([45], p. 76)

This recommendation concerns visual analytics which is not equivalent to Infovis, but the two clearly share much in common and the motivation for this call can equally be applied to Infovis.

While we believe that few would argue with the merits of the goals in the recommendation, precisely defining what is being called for is not so easy. What does it mean to create a "science of interaction" in visual analytics and Infovis? The recommendation speaks of developing a taxonomy of interaction techniques and identifying under-explored areas for future research. These are noble efforts, but we believe that a science of interaction also should involve gaining a deeper understanding of the utility and value of interaction in these fields. What does interaction contribute to the analytic process?

For that matter, we might raise questions about the nature of interaction itself. In the context of Infovis, what is interaction and interactive behavior? Operations such as moving a dynamic query slider [3] to narrow the set of data points being shown or selecting an alternate point in a fisheye view [19] to change the focus seem like clear examples of interactive behavior. But consider a system where the user selects a menu operation to change from a scatter plot to a parallel coordinates of the data. Is that interaction?

The purpose of this article relates to the recommendation from *Illuminating the Path* that was discussed above. Defining a science of interaction is a lofty goal and we do not purport to do so here, but we do seek to take some initial steps toward that goal. Our objective is to further current understandings of the role that interaction plays

- Ji Soo Yi is with Health Systems Institute & H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, E-Mail: jisoo.yi@hsi.gatech.edu.
- Youn ah Kang and John T. Stasko are with School of Interactive Computing & GVU Center, Georgia Institute of Technology, E-Mail: ykang3@mail.gatech.edu and stasko@cc.gatech.edu.
- Julie A. Jacko is with Health Systems Institute, The Wallace H. Coulter Department of Biomedical Engineering, & College of Computing, Georgia Institute of Technology & Emory University, E-Mail: jacko@hsi.gatech.edu

Manuscript received 31 March 2007; accepted 1 August 2007; posted online 27 October 2007. Published 14 September 2007.

For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

in Infovis. More specifically, we seek to identify the fundamental ways that interaction is used in Infovis systems and the benefits it provides to them (and to users).

In the next section, we review prior research on interaction in Infovis and examine how other researchers have defined and characterized its virtues. In Section 3, we describe the research methods used to survey and analyze these interaction techniques. In Section 4, we describe the results of an extensive analytic investigation of interaction techniques and introduce seven fundamental ways that interaction contributes to the explorations and analyses people perform while using Infovis systems.

## 2 BACKGROUND

It seems appropriate to start a discussion of interaction in Infovis with a definition of the term, but in fact, finding a solid definition of interaction is challenging. In the broader context of HCI, Dix et al. simply describe interaction as “the communication between user and the system” (p. 124) [16]. Becker, Cleveland, and Wilks compactly define interaction as direction manipulation and instantaneous change [6]. Since interaction can occur even with a static image as described previously [38], interaction is certainly not a tangible concept. That is likely why Beaudouin-Lafon mentions that “HCI research is far from having solid (and falsifiable) theories of interaction” (p. 16) [5].

Nonetheless, interaction techniques are less difficult to define and are more tangible concepts than interaction itself. A static image does not have an associated interaction technique even though users can interact with it. Foley et al. define an interaction technique as a way of using a physical input/output device to perform a generic task in a human-computer dialogue [18].

The definition of interaction techniques in the context of Infovis should extend Foley’s definition, however, which was grounded in the general context of HCI. As Ware identifies via the phrase, “asymmetry in data rates” (p.382) [51], the amount of data flowing from Infovis systems to users is far greater than from users to systems. Thus, interaction techniques in Infovis seem more designed for changing and adjusting visual representation than for entering data into systems, which clearly is an important aspect of interaction in HCI.

We view interaction techniques in Infovis as the features that provide users with the ability to directly or indirectly manipulate and interpret representations. According to this view, a static image or an autonomously animated representation does not have associated interaction techniques. However, a menu interface for changing from a scatter plot to a parallel coordinates view is an interaction technique since it allows users to manipulate a representation even though it may be less interactive or direct. (Here and throughout this paper, we intentionally use the term “user” rather than “viewer” or “people” to emphasize the fact that users actively use and interact with Infovis systems.)

Taxonomies of interaction techniques would be helpful to achieve a better understanding of the design space of interaction. Table 1 summarizes several studies in Infovis proposing taxonomies that we think are relevant to the examination of interaction techniques. Even though many of the studies share common units, the taxonomies have significantly different levels of granularity. Some try to categorize low-level interaction techniques (e.g., [9, 12, 15, 24, 37, 54]); some provide dimensions to describe interaction techniques (e.g., [38, 47]); another moves past the low-level interaction techniques to provide a broader view of interaction including notions such as interaction spaces and parameters (e.g., [50]); while others focus more on users’ tasks (e.g., [4, 56]). This divergence suggests that there may be multiple ways or granularities to describe interaction techniques, which is also in line with Norman’s action cycle [30] that describes interaction between a user and the world using multiple steps (i.e., forming the goal, forming the intention, specifying an action, executing the action, perceiving the state of the world, interpreting the state of the world, and evaluating the

outcome). Also, this divergence implies that defining a comprehensive taxonomy is challenging. Since Infovis is still a growing field, it is highly possible that an interaction technique developed in the future will not be clearly categorized by one of the low-level interaction technique taxonomies.

Table 1. Infovis Taxonomies Relevant to Interaction Techniques

Publications	Taxonomic units
<i>Taxonomies of low-level interaction techniques</i>	
Shneiderman (1996) [37]	Overview, zoom, filter, details-on-demand, relate, history, and extract
Buja, Cook, and Swayne (1996) [9]	Focusing (choice of [projection, aspect ratio, zoom, pan], choice of [variable, order, scale, scale-aspect ratio, animation, and 3-D rotation]), linking (brushing as conditioning / sectioning / database query), and arranging views (scatter plot matrix and conditional plot)
Chuah and Roth (1996) [13]	Basic visualization interaction (BVI) operations: graphical operations (encode data, set graphical value, manipulate objects), set operations (create set, delete set, summarize set, other), and data operations (add, delete, derived attributes, other)
Dix and Ellis (1998) [15]	Highlighting and focus, accessing extra information – drill down and hyperlinks, overview and context, same representation / changing parameters, same data / changing representation, linking representation – temporal fusion
Keim (2002) [24]	Dynamic projections, interactive filtering, interactive zooming, interactive distortion, interactive linking and brushing
Wilkinson (2005) [54]	Filtering (categorical/continuous/multiple/fast filtering), navigating (zooming/panning/lens), manipulating (node dragging/categorical reordering), brushing and linking (brush shapes/brush logic/fast brushing), animating (frame animation), rotating, transforming (specification/assembly/display/tap/2 taps/3 taps)
<i>Taxonomical dimensions of interaction techniques</i>	
Tweedie (1997) [47]	Interaction types (manual, mechanized, instructable, steerable, and automatic) and directness (direct and indirect manipulation)
Spence (2007) [38]	Interaction modes (continuous, stepped, passive, and composite interaction)
<i>A taxonomy of interaction operations</i>	
Ward and Yang (2004) [50]	interaction operators (navigation, selection, distortion), interaction spaces (screen-space, data value-spaces, data structure-space, attribute-space, object-space, and visualization structure-space), and interaction parameters (focus, extents, transformation, and blender)
<i>Taxonomies of user tasks</i>	
Zhou and Feiner (1998) [56]	Relational visual tasks (associate, background, categorize, cluster, compare, correlate, distinguish, emphasize, generalize, identify, locate, rank, reveal, switch) and direct visual organizing and encoding tasks (encode)
Amar, Eagan, and Stasko (2005) [4]	Retrieve value, filter, compute derived value, find extremum, sort, determine range, characterize distribution, find anomalies, cluster, and correlate

While these taxonomies are certainly useful for better understanding interaction, to us they still lack something important. The first three sets focus strongly on interaction techniques and are relatively system-centric. The last set focuses on user goals without a main focus on interaction. We believe it would be beneficial to bridge these two efforts—to connect user objectives with the interaction techniques that help accomplish them.

Finally, measuring the effectiveness of a taxonomy is difficult itself. We are drawn to a discussion of this issue by Beaudouin-Lafon [5] who proposes three dimensions to evaluate interaction models: 1) descriptive power, “the ability to describe a significant

range of existing interface”; 2) evaluative power: “the ability to help assess multiple design alternatives”; and 3) generative power: “the ability to help designers create new designs” (p. 17). None of the taxonomies listed above appear to provide all three levels.

### 3 METHODS

In order to more systematically understand the underlying mechanisms of interaction, we began this research with the goal of building a comprehensive list of Infovis interaction techniques. Since it clearly would not be possible to examine all existing systems and techniques, we decided instead to review existing literature and Infovis systems as follows.

We began by reviewing existing literature containing taxonomies of Infovis interaction techniques, as mentioned in the Background section just above. Next, we examined a number of commercial Infovis systems (e.g., SeeIT by ADVISOR Solutions, Inc. (formerly Visual Insights) [1], Spotfire<sup>®</sup> by Spotfire, Inc. [2, 41], TableLens<sup>™</sup> by Inxight Software, Inc. [23, 33], and InfoZoom<sup>®</sup> by humanIT [22, 40]) since they, as general purpose Infovis tools, tend to have a broad set of multiple interaction techniques. We also reviewed articles introducing new Infovis interaction techniques (e.g., pan & zoom, overview & details, focus + context, and filter). Finally, we selected well-known papers in sub-areas of Infovis (e.g., multivariate, time-series, hierarchical, software, security, geographic, and social visualization) to cover various application areas. In total, we surveyed 59 papers and 51 systems and collected 311 individual interaction techniques actually implemented in Infovis systems.

Even though the list of interaction techniques was growing larger and larger, the efforts left us somewhat unsatisfied. It was not clear how useful this list of techniques would be or more importantly, whether it would be descriptive and meaningful. As the gathering process progressed, however, we began to notice common sets of techniques emerging and some styles of interaction being listed more frequently.

Accordingly, we decided to aggregate and cluster the different techniques by using an affinity diagramming method. We grouped similar interaction techniques and iteratively refined the groups according to the core concepts. During the grouping process, several competing grouping schemes emerged. Initial groups tended to be commonly-used interaction techniques in different Infovis systems, which were not that different from existing taxonomies of low-level interaction techniques. However, we soon found that these grouping schemes could not be robust because there were numerous variants of interaction techniques that did not fall into any commonly used interaction technique. We realized that for different representation techniques, different interaction techniques are used to perform a similar task or achieve a similar goal. For example, suppose that a user is exploring the relationship between two particular variables. In a scatter plot style of visualization as in Spotfire [2], this goal is achieved by designating the two variables to be plotted on the x and y axes. In TableLens [33], however, the goal can be achieved by positioning the two variables next to each other and sorting values with respect to one of the variables.

Thus, we turned our attention to what users achieve by using the interaction techniques rather than how the techniques provided by Infovis systems work. In doing so, we realized that many different styles of interaction techniques serve a relatively small set of purposes. For example, unfolding sub-categories in an interactive pie chart [15], drill-down in a treemap [36], and semantic zooming [32] all may appear very different, but we argue that they serve the same purpose, getting more details.

After several iterations of clustering the techniques, the notion of aggregating them by the user’s *intent* in performing an interactive operation began to emerge. We found that the concept of ‘What a user wants to achieve’, herein described as “user intent,” is quite effective to classify the low-level interaction techniques into a small number of descriptive high-level categories.

## 4 CATEGORIES

Based on the notion of user intent, the following seven categories of interaction in Infovis emerged from our study. Each category will be discussed in more detail in a subsequent sub-section. To each category, as a title, we assigned a short identifying name (e.g., *Select*) and also an illustrative phrase that captures the essence of the user’s intent in performing the interaction. We describe each category to provide a definition of what it means and we also include exemplary individual interaction techniques that fall within that category.

- *Select*: mark something as interesting
- *Explore*: show me something else
- *Reconfigure*: show me a different arrangement
- *Encode*: show me a different representation
- *Abstract/Elaborate*: show me more or less detail
- *Filter*: show me something conditionally
- *Connect*: show me related items

### 4.1 Select: mark something as interesting

*Select* interaction techniques provide users with the ability to mark a data item(s) of interest to keep track of it. When too many data items are presented on a view, or when representations are changed, it is difficult for users to follow items of interest. By making items of interest visually distinctive, users can easily keep track of them even in a large data set and/or with changes in representations.

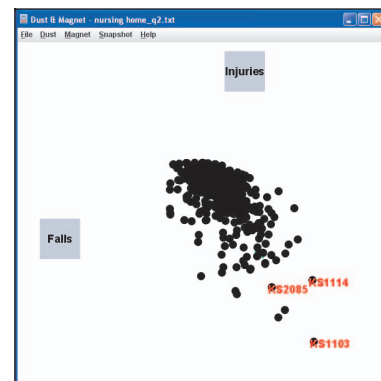


Fig. 1. A screen shot of Dust & Magnet showing the marking feature.

As shown in Fig. 1, the marking feature in Dust & Magnet [55], which visualizes data items as specks of iron that move when magnets (attributes) are manipulated, is an example of *Select*. With this technique, users can mark data items, and the marked items (KS1114, KS2085, and KS1103 in the figure) are labeled in red, so even after rearranging items, users can easily track and identify the location of items of interest. The spotlight feature in TableLens [33], a system that visualizes numerical data using bar charts in a tabular view, is a similar interaction technique except that Spotlight highlights data items instead of labeling. Yet another example of *Select* is the placemark feature in Google Earth [20], an interactive 3D geographic visualization tool. By putting a placemark on a location of interest, users can return to the location easily.

Interestingly, *Select* interaction techniques seem to work as a preceding action to subsequent operations. As shown in the Dust & Magnet and TableLens examples, users select data items of interest before rearranging, so that they can see where the items of interest would be located in the new arrangement. Rather than acting as a standalone technique, *Select* interaction is coupled with other interaction techniques to enrich user exploration and discovery.

### 4.2 Explore: show me something else

*Explore* interaction techniques enable users to examine a different subset of data cases. When users view data using an Infovis system, they often can only see a limited number of data items at a time because of some combination of the large scale of the data set, view

and/or screen limitations, and fundamental perceptual and cognitive limitations in human information processing. Infovis system users typically examine a subset of the data to gain understanding and insight, and then they move on to view some other data. *Explore* interactions do not necessarily make complete changes in the data being viewed, however. More frequently, some new data items enter the view as others are removed.

The most common *Explore* interaction technique in our survey is panning. Panning refers to the movement of a camera across a scene or scene movement while the camera stays still. Panning is often achieved by a special mode where the user grabs the scene and moves it with a mouse or by simply altering the view via scrollbars. Many Infovis systems use panning techniques: for example, Spotfire [2], Vizster [21], Dust & Magnet [55], and SeeIT [1].

Another example of an *Explore* interaction is the Direct-Walk technique. Direct-Walk allows users to smoothly move the viewing focus from one position in information structure to another by “a series of mouse points or other direct-manipulation methods” (p. 239) [11]. The hyperlink feature in the Jazz zooming interface toolkit [7] is an example of Direct-Walk. Hyperlinks move the user from one point in the information space to another in a smooth, animated transition. An online graphical dictionary, Visual Thesaurus® [44], is another example of Direct-Walk. In Visual Thesaurus®, a searched vocabulary is displayed at the center surrounded by related vocabularies as shown Fig. 2. When one of surrounding words is clicked, the word smoothly comes to the center and new related words surround this newly centered word.

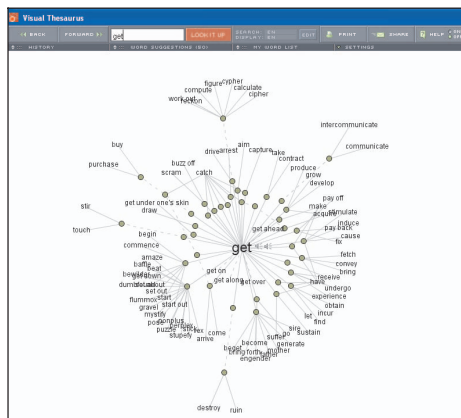


Fig. 2. A screen shot of Visual Thesaurus® with the word “get” at the center

### 4.3 Reconfigure: show me a different arrangement

*Reconfigure* interaction techniques provide users with different perspectives onto the data set by changing the spatial arrangement of representations. One of the essential purposes of Infovis is to reveal hidden characteristics of data and the relationships between them. A good static representation often serves this purpose, but a single representation rarely provides sufficient perspectives. Thus, many Infovis tools incorporate *Reconfigure* interaction techniques that allow users to change the way data items are arranged or the alignment of data items in order to provide different perspectives on the data set.

The sorting and rearranging columns operations in TableLens [33] are good examples of *Reconfigure* techniques. As shown in Fig. 3, by sorting the “Horsepower” column, users can determine that horsepower values of vehicles are roughly correlated with cylinders, displacement, and weight. Also, users can rearrange the columns to compare attributes of interest side by side. Sorting and rearranging columns (or rows) features can be found in other Infovis systems containing tabular views as well, such as InfoZoom [40].

The capability of changing the attributes presented on the axes in a scatter plot view of Spotfire [2] is a similar, but different example of a *Reconfigure* technique. Changing the attributes assigned to x-

and y-axes changes the sets of attributes or variables to be examined among the entire data set, so it eventually changes relationships between data items and provides different perspectives.

The baseline adjustment feature in a stacked histogram, as shown in Fig. 4, enables users to better compare the heights of subsections of the histogram [15]. Without this technique, it is difficult to compare the values of subsections (values of the West variable in the figure) not initially on the bottom of the histogram. The Selective Dynamic Manipulation (SDM) system [14], which introduced many interaction techniques for 2D and 3D visualization, provides a similar technique to compare the heights of bars in three-dimensional visualization. Since a distant object appears smaller than a nearby object in a 3D view, comparing the two objects’ heights, for example, is challenging. SDM allows users to bring objects in a 3D view to a front 2D plane with a common baseline so that users can compare the sizes more accurately.

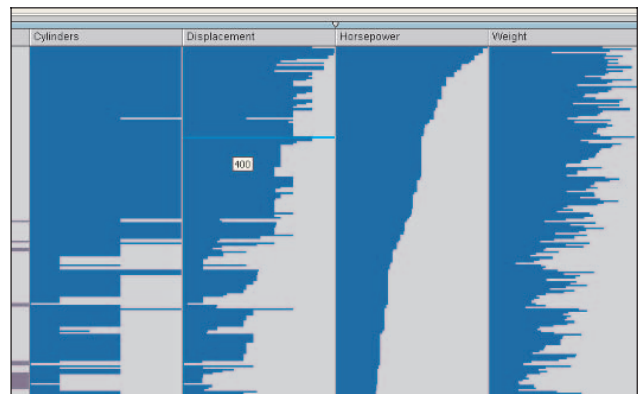


Fig. 3. A screen shot of TableLens using the sort function on the “Horsepower” column

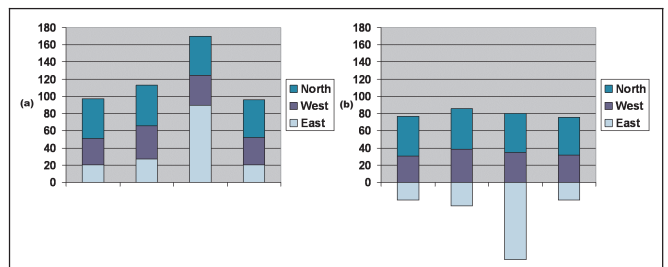


Fig. 4. Stacked histograms: (a) an original view and (b) a view with baseline adjustment

Other system’s interaction techniques allow users to move data items more freely to make the arrangement more suitable for their mental model. For example, users of the online social network visualization system Vizster [21] can move nodes freely and thus can arbitrarily cluster a certain set of people (e.g., family, friends, and business contacts). The Data Mountain [34] that presents web browser favorites as thumbnails on an inclined plane is another similar example as it allows users to arrange groups of related web pages at various positions on the plane.

*Reconfigure* techniques also include a set of interaction techniques reducing occlusions. Since many Infovis systems present large amounts of data, individual data cases often visually overlap. Especially in 3D representation techniques, distant data items are often occluded by nearby data items in the same line of sight.

For example, the view rotation operation in many 3D Infovis systems (e.g., SDM [14]) helps reduce occlusion in a 3D visualization. Such a feature helps users rotate their line of sight to see through a cloud of data items. A similar, but slightly different example is a technique in ConeTrees [35] where users rotate a portion of the tree instead of rotating the line of sight in order to see occluded data items.



Another example interaction in this category is the jitter operation as implemented in systems like Spotfire [2]. When many data cases are drawn to particular vertical or horizontal rows, items may overlap resulting in occlusion. By applying jitter, the position of each item is randomly shifted by a small spatial increment, thus uncovering many more items and providing a better sense of the density of items in a region. Fig. 5 illustrates the results of jitter in Spotfire. A similar technique in Dust & Magnet is the “Spread Dust” operation that makes data items (dust particles) gradually repel each other so that occlusion decreases [55].

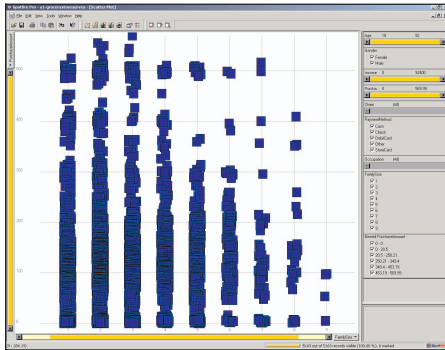


Fig. 5. A screen shot of Spotfire showing the result of the jitter technique.

#### 4.4 Encode: show me a different representation

*Encode* techniques enable users to alter the fundamental visual representation of the data including visual appearance (e.g., color, size, and shape) of each data element. In Infovis systems, visual elements serve an important role not only because they can affect pre-attentive cognition but also because they are directly related to how users understand relationships and distributions of the data items. For instance, by encoding height information to a map using a spectrum of color, users can better identify the height information (e.g., the height of a mountain) without altering the spatial arrangement of the map.

Simply changing how the data is represented (e.g., changing a pie chart to a histogram) is an example of *Encode*. By changing a type of representation, users expect to uncover new aspects of relationship. Infovis systems that provide multiple representations of data, for example, Spotfire [2] and Xmdv tool [49], have this capability.

Another widely used technique of *Encode* is the set of interaction techniques that alter the color encoding of a data set. Many Infovis techniques (e.g., Dust & Magnet [55], InfoScope by Macrofocus [27], and Spotfire [2]) enable users to adjust a color or a spectrum of colors for a certain variable. Since color encoding is changed instantly and dynamically, users can experiment with various color encoding schemes to find the most suitable one. Additive color encoding in Attribute Explorer [39] is an advanced color encoding technique, which helps users understand distributions of multiple variables rather than a single variable.

Beyond color encoding, many systems provide other encoding techniques, such as size (e.g., Dust & Magnet [55]), orientation (e.g., Polaris [43]), font (e.g., SemaSpace [31]), and shape (e.g., Spotfire [2]). Since some of encoding techniques can be used simultaneously, they are often used together to encode many variables into representation. Again, interactivity is essential to help users find a proper encoding scheme.

#### 4.5 Abstract/Elaborate: show me more or less detail

*Abstract/Elaborate* interaction techniques provide users with the ability to adjust the level of abstraction of a data representation. These types of interactions allow users to alter the representation from an overview down to details of individual data cases and often many levels in-between. The user’s intent correspondingly varies

between seeking more of a broad, contextual view of the data to examining the individual attributes of a data case or cases.

An exemplary interaction technique in this category is any technique from the set of details-on-demand operations. For example, the drill-down operation in a treemap visualization, such as SequoiaView (formerly known Cushion Tree [48]), allows a user to examine a particular sub-tree within an information hierarchy. Similarly, the animated details-on-demand techniques of SunBurst [42] (i.e., angular detail, detail inside, and detail outside) provide very similar functionality by allowing particular sub-trees in a hierarchy to be examined more closely without losing context of the entire structure. TableLens [33] also allows users to focus on a data case and its details (text of actual values) emerge. Furthermore, simple tool-tip interaction techniques that provide detailed information when a mouse cursor hovers over a data item also belong to this category: for example, SeeIT [1] as shown in Fig. 6.

Another very common but slightly complex example of *Abstract/Elaborate* techniques is zooming (or geometric zooming if it is to be distinguished from semantic zooming). Through zooming, users can simply change the scale of a representation so that they can see an overview of a larger data set (using zoom-out) or the detailed view of a smaller data set (using zoom-in). A key point here is that the representation is not fundamentally altered during zooming. Details simply come more clearly into focus or fade away into context.

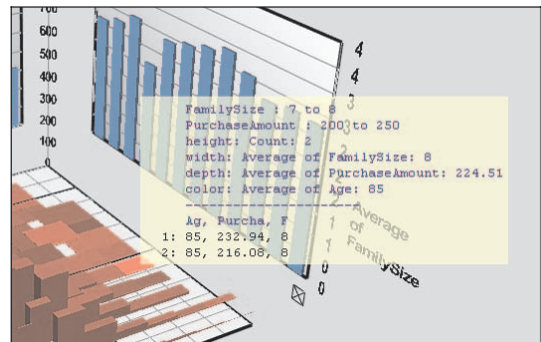


Fig. 6. A screen shot of SeeIT showing the tool tip feature

#### 4.6 Filter: show me something conditionally

*Filter* interaction techniques enable users to change the set of data items being presented based on some specific conditions. In this type of interaction, users specify a range or condition, so that only data items meeting those criteria are presented. Data items outside of the range or not satisfying the condition are hidden from the display or shown differently, but the actual data usually remain unchanged so that whenever users reset the criteria, the hidden or differently shown data items can be recovered. The user is not changing perspective on the data, just specifying conditions on which data are shown.

Dynamic query controls [3] as used in many Infovis systems (e.g., Spotfire [2]) are a representative example of this type of interaction. Users select ranges by moving sliders or particular values by clicking on check boxes and the data cases meeting those constraints are immediately shown. This type of interaction helps make a system feel much more responsive and live as compared to traditional batch-oriented text queries. Variants of dynamic query controls such as alphaliders, rangsliders, and toggle buttons are used to filter textual data, numerical data, and categorical data, respectively.

The Attribute Explorer [39] extends dynamic query capabilities by changing the colors of filtered data items rather than removing them from the display, as shown in Fig. 7. This helps users understand the context of the dataset by showing nearby data items not quite meeting the filtering criteria.

The Name Voyager [53], a website that illustrates the popularity of baby names over time, also supports a filtering interaction. Instead of using specific controls, users can filter the data items (e.g., names) through keyboard interaction. For example, as shown in Fig. 8, when

a user types “K”, only baby names starting with K are shown on the display. If the user types “I” following K, the system filters the data set and only shows names that start with “KI”. By this simple and intuitive interaction technique, Name Voyager provides a very natural visual exploration of the data. QuerySketch [52] is yet another interesting example of *Filter* techniques. QuerySketch allows users to draw a line graph freehand, and then the system retrieves and presents data cases with similar graphs. These graphs frequently represent time series data.

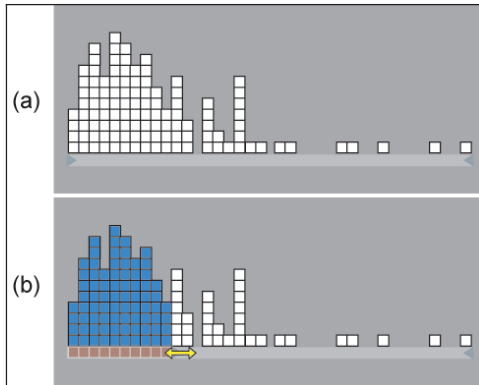


Fig. 7. Attribute Explorer style display: (a) before changing limits and (b) after changing the lower limit

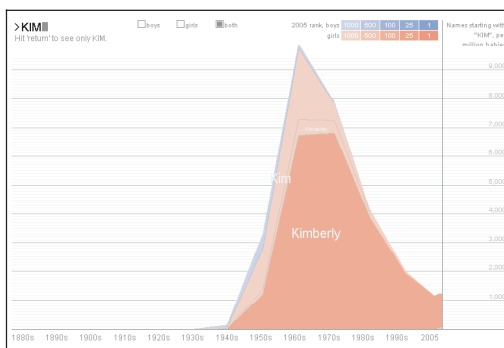


Fig. 8. A screen shot of Name Voyager showing names with “KIM”

**4.7 Connect: show me related items**

*Connect* refers to interaction techniques that are used to (1) highlight associations and relationships between data items that are already represented and (2) show hidden data items that are relevant to a specified item.

When multiple views are used to show different representations of the same data set (e.g., 3D scatter plot and 2D scatter plot as shown in Fig. 9), it may be difficult to identify the corresponding item for a data case in other view(s). To alleviate this difficulty, the brushing technique is used to highlight the representation of a selected data item in the other views being displayed. In Fig. 9, when a user selects a data item in the left view, the same data item of the right view is highlighted (circled in this case) simultaneously.

*Connect* interactions can apply to situations involving a single view as well. For example, in Vizster [21], hovering a mouse cursor over a node highlights directly connected nodes (friends) or neighbors of directly connected nodes (friends of friends). Here, the connection is not to other representations of the same item as in brushing but to items that harbor relationship to a focus element.

*Connect* interaction techniques also reveal related data items which are originally not shown. In Vizster, double clicking a node causes expansion of the node, so that the related nodes for the focus node (the person) are added. A similar but different example is the aforementioned Visual Thesaurus® [44], where clicking a word in a view reveals related words, and other unrelated words in the original view disappear. Keen readers might notice that this interaction

technique was already categorized as *Explore*, which will be discussed more in the Discussion section.

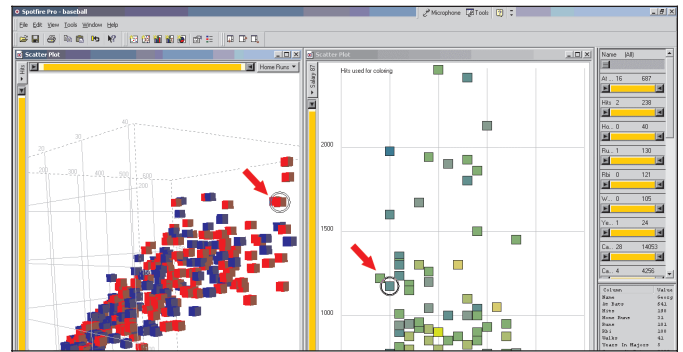


Fig. 9. A screen shot of Spotfire showing a brushing technique

**4.8 Other Interaction Techniques**

Other interaction techniques in Infovis systems certainly exist. For instance, consider a broad set of operations found commonly in many interactive applications. A few examples are listed below:

- Undo/redo: techniques that allow users to go backward or forward to pre-existing system states (e.g., undo, redo, history, and reset)
- Change configuration: techniques that allow users to change various configurations and settings of a system (e.g., change locations of dynamic queries in Spotfire [2])

Because these operations are common to many different types of applications and are not unique to Infovis, we have chosen not to include them in our scheme. This, however, does not diminish their value as useful interactive capabilities in information visualization.

**5 DISCUSSION**

It is difficult to create categories of interaction techniques that are clear and comprehensive. The categories we proposed are based on our own perspective on interaction in Infovis and, thus, inherently debatable. In this section, we discuss issues with our categorization and the nature of interaction in Infovis.

Through the categorization process, we realized that the categories are not collectively exhaustive. Some techniques are difficult to classify and do not quite fit into any one of the categories. For example, as shown in Fig. 10, the water level technique in SeeIT [33], which visualizes multivariate data in a 3D view with projection walls, provides a movable baseline that can be adjusted up and down, so that users can compare the heights of 3D histograms. Here, users do not interact directly with data items and the representation of data items remains unchanged. It is a technique that adds a layer on top of the representation and plays a role as a cognitive aid to augment a user’s ability to compare the values. This interaction simply did not seem to fit well into any of our categories.

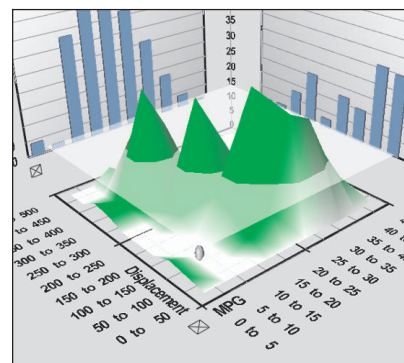


Fig. 10. A screen shot of SeeIT showing the water level feature

Some other interaction techniques appear to fulfill multiple user intents, which make it possible to classify them into several categories. For example, semantic zooming [32] is not only an *Elaborate/Abstract* technique due to its zooming capability, but also an *Encode* technique since the data representation can change as the zooming scale changes. The Magic Lens [17], which provides a different sub view on top of the main view using a lens metaphor, is another good example of having multiple intents. One popular usage of Magic Lens is using it as movable and stackable filter. However, it can also provide many other functions (e.g., color encoding, changing representation, zooming, and providing details), which makes it difficult to classify into a single category. The aforementioned interaction technique in Visual Thesaurus<sup>®</sup> [44] is another example, which is categorized as both *Connect* and *Explore* since users can explore new words by clicking a connected word.

We also considered other categories to be added to our final set. For example, we debated adding a *Compare* category but ultimately omitted it because we believe that *Compare* is a higher-level user goal or objective than the other user intents we identify. *Compare* simply can mean so many different things. In particular, the intents we identify often make up components of a broader comparison goal—"Let me *Filter* data so that I can compare items of interest"; "Let me *Reconfigure* to compare these two subsets more easily"; and "Let me *Encode* variable A to easily compare this attribute."

Despite all these debatable issues and exceptions, we still believe that this categorization, based on user intents, is a useful approach and has several strengths. In order to assess its utility in a systematic manner, we use Beaudouin-Lafon's three dimensions (i.e., descriptive, evaluative, and generative power) for evaluating interaction models. First, we believe that our user-intent-centric categorization has fairly good descriptive power in that it captures the characteristics of interaction techniques at a higher level. While a simple enumeration of interaction techniques often fails to embrace variants of existing techniques or new techniques, our categories are less vulnerable to new developments as long as they serve one of user intentions we have identified. Second, our categories can help designers and developers to examine whether users' needs are fulfilled by a system, which implies that it has an evaluative power to some degree. Understanding what is missing in a system in terms of supporting user intents could be a more meaningful way to evaluate the system than simply checking whether the system has a particular feature that is commonly used in other systems. Finally, we argue that it also has generative power. Even though the categorization may not directly help Infovis designers generate new ideas of interaction techniques, it provides at least some common vocabularies to think about different users' intentions when developing new techniques. Since understanding what users need promotes the creativity of designers [26], we believe that our categories may have contributions in that aspect.

We began this article with a statement that Infovis systems appear to have two different fundamental components: representation and interaction. Through our work and analysis, however, we came to a conclusion that maybe it is not so easy to separate the two. On the one hand, some may argue that interaction is all about representation in that it plays a role merely as an operator that changes representation. Without interaction, however, representation is no more than a static image. By supporting further exploration of data items, interaction enables users to have multiple perspectives and gain insight on the data set. It is what separates an Infovis system from a static image. We conclude that these two components are in a symbiotic relationship.

One way to distinguish representation and interaction might be through temporal characteristics. While representation is not dependent on time per se, interaction fundamentally involves changes over time. A basic tradeoff exists between the time to perform interaction activities (e.g., generating a different view) and the space required to present multiple static images (e.g., screen real estate). To achieve the same variety of representations without interaction, one would need a huge display. Thus, as can be seen,

there is a tradeoff between using multiple static representations and one, interactive representation.

Nonetheless, the value of representation and interaction in helping users understand information, while reducing cognitive burden, makes it impossible to separate the two. More research is needed to better understand how to leverage each component to build optimal Infovis systems.

## 6 CONCLUSIONS

In this paper, we proposed seven different categories of interaction techniques based on user intents. We believe that this article makes two main contributions to the Infovis domain.

First, our efforts draw attention to the importance of interaction in Infovis research and reveal its subtle complexity. While existing research in the area often focuses on representation, we highlight the overshadowed, but very important interaction component and strongly argue that it provides a way to overcome the limits of representation and augment a user's cognition.

Second, we provide a novel user intent-based categorization to discuss and characterize interaction techniques in Infovis. In conjunction with other interaction taxonomies, our categories might be able to provide a bigger picture view of interaction. For example, using these categories, it would be useful and meaningful to discuss what type of user intent a system supports (or not) as well as what tasks a system supports (or not).

Certainly, simply having these categories is far from our eventual goal of establishing the science of interaction in Infovis. However, we believe that these categories are an initial step toward this direction. We believe that this categorization better articulates the ways in which interaction techniques are used, while providing a more useful common vocabulary (of user intent) for further discussion and application in the development of Infovis systems. Our categorization, coupled with an exhaustive list of interaction techniques as well as higher-level user tasks, would provide a holistic framework that moves closer to providing a true science of interaction.

## ACKNOWLEDGEMENTS

This research is supported in part by the National Science Foundation via Award IIS-0414667 and the National Visualization and Analytics Center (NVAC<sup>™</sup>), a U.S. Department of Homeland Security Program, under the auspices of the SouthEast Regional Visualization and Analytics Center. It is also supported in part by GVU Center Seed Grant. The authors also wish to thank Erin Kinzel and Kevin Moloney for their careful review of this document.

## REFERENCES

- [1] ADVISOR Solutions Inc., "SeeIT," <http://www.advizorsolutions.com/>, 2007.
- [2] C. Ahlberg, "Spotfire: an information exploration environment," *SIGMOD Record*, vol. 25, pp. 25-29, 1996.
- [3] C. Ahlberg, C. Williamson, and B. Shneiderman, "Dynamic Queries for Information Exploration: An Implementation and Evaluation," presented at Conference on Human Factors in Computing Systems (CHI '92), Monterey, CA, USA, pp. 619-626, 1992.
- [4] R. Amar, J. Eagan, and J. T. Stasko, "Low-Level Components of Analytic Activity in Information Visualization," presented at IEEE Symposium on Information Visualization (InfoVis '05), pp. 111-117, 2005.
- [5] M. Beaudouin-Lafon, "Designing interaction, not interfaces," presented at the working conference on Advanced visual interfaces (AVI '04), Gallipoli (LE), Italy, pp. 15-22, 2004.
- [6] R. A. Becker, W. S. Cleveland, and A. R. Wilks, "Dynamic Graphics for Data Analysis," *Statistical Science*, vol. 2, pp. 355-383, 1987.
- [7] B. B. Bederson, J. Grosjean, and J. Meyer, "Toolkit design for interactive structured graphics," *IEEE Transactions on Software Engineering*, vol. 30, pp. 535-546, 2004.

- [8] J. Bertin, *Semiology of graphics*: University of Wisconsin Press, 1983.
- [9] A. Buja, D. Cook, and D. F. Swayne, "Interactive High-Dimensional Data Visualization," *Journal of Computational and Graphical Statistics*, vol. 5, pp. 78-99, 1996.
- [10] A. Buja, J. A. McDonald, J. Michalak, and W. Stuetzle, "Interactive data visualization using focusing and linking," presented at IEEE Conference on Visualization (Visualization '91), San Diego, California, pp. 156-163, 1991.
- [11] S. K. Card, P. Pirolli, and J. D. Mackinlay, "The cost-of-knowledge characteristic function: display evaluation for direct-walk dynamic information visualizations," ACM Press New York, NY, USA, 1994, pp. 238-244.
- [12] M. C. Chuah and S. F. Roth, "On the Semantics of Interactive Visualizations," presented at IEEE Symposium on Information Visualization (InfoVis '96), San Francisco, CA, USA, pp. 29-36, 1996.
- [13] M. C. Chuah and S. F. Roth, "On the Semantics of Interactive Visualizations," presented at IEEE Symposium on Information Visualization, San Francisco, CA, pp. 29-36, 1996.
- [14] M. C. Chuah, S. F. Roth, J. Mattis, and J. Kolojechick, "SDM: selective dynamic manipulation of visualizations," presented at ACM symposium on User interface and software technology (UIST '95), pp. 61-70, 1995.
- [15] A. Dix and G. Ellis, "Starting simple: adding value to static visualisation through simple interaction," presented at the working conference on Advanced visual interfaces (AVI '98), L'Aquila, Italy, pp. 124-134, 1998.
- [16] A. Dix, J. Finlay, G. D. Abowd, and R. Beale, *Human-computer interaction*, 3rd ed: Pearson Prentice Hall, 2004.
- [17] K. Fishkin and M. C. Stone, "Enhanced dynamic queries via movable filters," presented at Conference on Human Factors in Computing Systems (CHI '95), Denver, CO, USA, pp. 415-420, 1995.
- [18] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics: Principles and Practice in C*, 2nd ed: Addison-Wesley Professional, 1995.
- [19] G. W. Furnas, "Generalized fisheye views," presented at Conference on Human Factors in Computing Systems (CHI '86), Boston, MA, USA, pp. 16-23, 1986.
- [20] Google Inc., "Google Earth," <http://earth.google.com/>.
- [21] J. Heer and D. Boyd, "Vizster: Visualizing Online Social Networks," presented at IEEE Symposium on Information Visualization (InfoVis '05), Minneapolis, MN, USA, pp. 33-40, 2005.
- [22] HumanIT, "InfoZoom," <http://www.infozoom.com/enu/index.htm>.
- [23] Inxight Software Inc., "Table Lens," <http://www.inxight.com/products/sdks/tl/>.
- [24] D. A. Keim, "Information Visualization and Visual Data Mining," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, pp. 1-8, 2002.
- [25] R. Kosara, H. Hauser, and D. Gresh, "An Interaction View on Information Visualization," presented at EUROGRAPHICS 2003 (EG '03), pp. 123-137, 2003.
- [26] B. Lawson, *How Designers Think: The Design Process Demystified*, 3rd ed: Architectural Press, 1997.
- [27] Macrofocus GmbH, "InfoScope," <http://www.macrofocus.com/public/products/infoscope.html>.
- [28] T. Miller and J. T. Stasko, "The InfoCanvas: information conveyance through personalized, expressive art," presented at Conference on Human Factors in Computing Systems (CHI '01), Seattle, WA, USA, pp. 305-306, 2001.
- [29] D. A. Norman, *Things that make us smart*. Reading, MA, USA: Addison-Wesley Pub. Co., 1993.
- [30] D. A. Norman, *The design of everyday things*: Basic Books, 2002.
- [31] D. Offenhuber and G. Dirmoser, "SemaSpace - Semantic Networks as Memory Theatre," <http://residence.aec.at/didi/FLweb/semaspace.pdf>.
- [32] K. Perlin and D. Fox, "Pad: an alternative approach to the computer interface," presented at Computer graphics and interactive techniques, pp. 57-64, 1993.
- [33] R. Rao and S. K. Card, "The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information," presented at Conference on Human Factors in Computing Systems (CHI '94), Boston, MA, USA, pp. 318 - 322, 1994
- [34] G. Robertson, M. Czerwinski, K. Larson, D. C. Robbins, D. Thiel, and M. van Dantzich, "Data mountain: using spatial memory for document management," presented at ACM symposium on User interface software and technology (UIST '98), San Francisco, CA, USA, pp. 153-162, 1998.
- [35] G. G. Robertson, J. D. Mackinlay, and S. K. Card, "Cone Trees: animated 3D visualizations of hierarchical information," presented at Conference on Human Factors in Computing Systems (CHI '91), New Orleans, LA, USA, pp. 189-194, 1991.
- [36] B. Shneiderman, "Tree visualization with tree-maps: 2-d space-filling approach," *ACM Transactions on Graphics*, vol. 11, pp. 92-99, 1992.
- [37] B. Shneiderman, "The eyes have it: a task by data type taxonomy for information visualizations," presented at IEEE Symposium on Visual Languages, 1996, Boulder, CO, USA, pp. 336-343, 1996.
- [38] R. Spence, *Information Visualization: Design for Interaction*, 2nd ed: Prentice Hall, 2007.
- [39] R. Spence and L. Tweedie, "The Attribute Explorer: information synthesis via exploration," *Interacting with Computers*, vol. 11, pp. 137-146, 1998.
- [40] M. Spenke and C. Beilken, "InfoZoom - Analysing Formula One racing results with an interactive data mining and visualisation tool," presented at International Conference on Data Mining, Cambridge University, United Kingdom, pp. 455-64, 2000.
- [41] Spotfire Inc., "Spotfire," <http://www.spotfire.com/>. 2007.
- [42] J. T. Stasko and E. Zhang, "Focus+ Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations," presented at IEEE Symposium on Information Visualization (InfoVis '00), Salt Lake City, UT, USA, pp. 57-65, 2000.
- [43] C. Stolte, D. Tang, and P. Hanrahan, "Polaris: a system for query, analysis, and visualization of multidimensional relational databases," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, pp. 52-65, 2002.
- [44] Thinkmap Inc., "Thinkmap Visual Thesaurus," <http://www.visualthesaurus.com/>.
- [45] J. J. Thomas and K. A. Cook, "Illuminating the Path." Los Alamitos, CA, USA: IEEE, 2005.
- [46] E. R. Tufte, *Envisioning information*: Graphics Press, 1990.
- [47] L. Tweedie, "Characterizing Interactive Externalizations," presented at Conference on Human Factors in Computing Systems (CHI '97), Atlanta, GA, pp. 375 - 382, 1997.
- [48] J. J. Van Wijk and H. Van de Wetering, "Cushion treemaps: visualization of hierarchical information," presented at IEEE Symposium on Information Visualization (InfoVis '99), San Francisco, CA, USA, pp. 73-78, 1999.
- [49] M. O. Ward, "XmdvTool: integrating multiple methods for visualizing multivariate data," presented at IEEE Conference on Visualization (Visualization '94), Washington, DC, USA, pp. 326-333, 1994.
- [50] M. O. Ward and J. Yang, "Interaction Spaces in Data and Information Visualization," presented at Joint Eurographics/IEEE TCVG Symposium on Visualization, Konstanz, Germany, pp. 137-145, 2004.
- [51] C. Ware, *Information Visualization: Perception for Design*. San Diego, CA, USA: Academic Press, 2000.
- [52] M. Wattenberg, "Sketching a graph to query a time-series database," presented at Conference on Human Factors in Computing Systems (CHI '01), Seattle, WA, USA, pp. 381-382, 2001.
- [53] M. Wattenberg and J. Kriss, "Designing for Social Data Analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, pp. 549-557, 2006.
- [54] L. Wilkinson, *The Grammar of Graphics*, 2nd ed. New York, NY, USA: Springer, 2005.
- [55] J. S. Yi, R. Melton, J. T. Stasko, and J. A. Jacko, "Dust & Magnet: multivariate information visualization using a magnet metaphor," *Information Visualization*, vol. 4, pp. 239-256, 2005.
- [56] M. X. Zhou and S. K. Feiner, "Visual task characterization for automated visual discourse synthesis," presented at Conference on Human Factors in Computing Systems (CHI '98), Los Angeles, CA, USA, pp. 392-399, 1998.

# Evaluating Information Visualizations

Sheelagh Carpendale

Department of Computer Science, University of Calgary,  
2500 University Dr. NW, Calgary, AB, Canada T2N 1N4  
sheelagh@ucalgary.ca

## 1 Introduction

Information visualization research is becoming more established, and as a result, it is becoming increasingly important that research in this field is validated. With the general increase in information visualization research there has also been an increase, albeit disproportionately small, in the amount of empirical work directly focused on information visualization. The purpose of this paper is to increase awareness of empirical research in general, of its relationship to information visualization in particular; to emphasize its importance; and to encourage thoughtful application of a greater variety of evaluative research methodologies in information visualization.

One reason that it may be important to discuss the evaluation of information visualization, in general, is that it has been suggested that current evaluations are not convincing enough to encourage widespread adoption of information visualization tools [57]. Reasons given include that information visualizations are often evaluated using small datasets, with university student participants, and using simple tasks. To encourage interest by potential adopters, information visualizations need to be tested with real users, real tasks, and also with large and complex datasets. For instance, it is not sufficient to know that an information visualization is usable with 100 data items if 20,000 is more likely to be the real-world case. Running evaluations with full data sets, domain specific tasks, and domain experts as participants will help develop much more concrete and realistic evidence of the effectiveness of a given information visualization. However, choosing such a realistic setting will make it difficult to get a large enough participant sample, to control for extraneous variables, or to get precise measurements. This makes it difficult to make definite statements or generalize from the results. Rather than looking to a single methodology to provide an answer, it will probably will take a variety of evaluative methodologies that together may start to approach the kind of answers sought.

The paper is organized as follows. Section 2 discusses the challenges in evaluating information visualizations. Section 3 outlines different types of evaluations and discusses the advantages and disadvantages of different empirical methodologies and the trade-offs among them. Section 4 focuses on empirical laboratory experiments and the generation of quantitative results. Section 5 discusses qualitative approaches and the different kinds of advantages offered by pursuing this type of empirical research. Section 6 concludes the paper.

## 2 Challenges in Evaluating Information Visualizations

Much has already been written about the challenges facing empirical research in information visualization [2, 12, 53, 57]. Many of these challenges are common to all empirical research. For example, in all empirical research it is difficult to pick the right focus and to ask the right questions. Given interesting questions, it is difficult to choose the right methodology and to be sufficiently rigorous in procedure and data collection. Given all of the above, appropriate data analysis is still difficult and perhaps most difficult of all is relating a new set of results to previous research and to existing theory. However, information visualization research is not alone in these difficulties; the many other research fields that also face these challenges can offer a wealth of pertinent experience and advice.

In particular, empirical research in information visualization relates to human computer interaction (HCI) empirical research, perceptual psychology empirical research, and cognitive reasoning empirical research. The relationship to empirical research in HCI is evident in that many of the tasks of interest are interface interaction tasks, such as zooming, filtering, and accessing data details [66]. The aspects of these interactive tasks that provide access to the visual representation and its underlying dataset often relate to the usability of a system. Other challenges that are shared with HCI empirical research include the difficulty of obtaining an appropriate sample of participants. If the visualization is intended for domain experts it can be hard to obtain their time. Also, when evaluating complex visualization software, it may not be clear whether the results are due to a particular underlying technique or the overall system solution. If an existing piece of software is to be used as a benchmark against which to compare an interactive visualization technique, it is likely that participants may be much more familiar with the existing software and that this may skew the results. This problem becomes more extreme the more novel a given visualization technique is. Research prototypes are not normally as smooth to operate as well established software, creating further possibilities for affecting study results and leading to controversy about testing research prototypes against the best competitive solution. Greenberg and Buxton [27] discuss this problem in terms of interaction sketches, encouraging caution when thinking about conducting usability testing on new ideas and new interface sketches in order to avoid interfering with the development of new ideas. In addition, research software does not often reach a stage in which it can support a full set of possible tasks or be fully deployable in real-world scenarios [57].

In addition to usability questions, perceptual and comprehensibility questions such as those considered in perceptual psychology are important in assessing the appropriateness of a representational encoding and the readability of visuals [30, 79]. Also, in information visualization, there are a great variety of cognitive reasoning tasks that vary with data type and character, from low-level detailed tasks to complex high-level tasks. Some of these tasks are not clearly defined, particularly those that hold some aspect of gaining new insight into the data, and may be more challenging to test empirically. Examples of low-level detailed tasks include such tasks as compare, contrast, associate, distinguish, rank, cluster, correlate, or categorize [57]; higher-level and more complex cognitive tasks include developing an understanding of data trends, uncertainties, causal relationships, predicting the future, or learning a domain [1]. Many important tasks can require weeks or months to complete. The success of information

visualization is often an interplay between an expert's meta-knowledge and knowledge of other sources as well as information from the visualization in use.

While all of the above are important, a question that lies at the heart of the success of a given information visualization is whether it sheds light on or promotes insight into the data [55, 63]. Often, the information processing and analysis tasks are complex and ill-defined, such as discovering the unexpected, and are often long term or on-going. What exactly insight is probably varies from person to person and instance to instance; thus it is hard to define, and consequently hard to measure. Plaisant [57] describes this challenge as "answering questions you didn't know you had." While it is possible to ask participants what they have learned about a dataset after use of an information visualization tool, it strongly depends on the participants' motivation, their previous knowledge about the domain, and their interest in the dataset [55, 63]. Development of insight is difficult to measure because in a realistic work setting it is not always possible to trace whether a successful discovery was made through the use of an information visualization since many factors might have played a role in the discovery. Insight is also temporally elusive in that insight triggered by a given visualization may occur hours, days, or even weeks after the actual interaction with the visualization. In addition, these information processing tasks frequently involve teamwork and include social factors, political considerations and external pressures such as in emergency response scenarios. However, there are other fields of research that are also grappling with doing empirical research in complex situations. In particular, ecologists are faced with conducting research towards increasing our understanding of complex adaptive systems. Considering the defining factors of complex adaptive systems may help to shed some light on the difficulties facing empirical research in information visualization. These factors include non-linearity, holoarchy and internal causality [37, 49]. When a system is non-linear, the system behaviour comes only from the whole system. That is, the system can not be understood by decomposing it into its component parts which are then reunited in some definitive way. When a system is holoarchical it is composed of holons which are both a whole and a part. That is, the system is mutually inter-nested. While it is not yet common to discuss information analysis processes in terms of mutual nesting, in practice many information analysis processes are mutually nested. For instance, consider the processes of search and verification: when in the midst of searching, one may well stop to verify a find; and during verification of a set of results, one may well need to revert to search again. Internal causality indicates that the system is self-organizing and can be characterized by goals, positive and negative feedback, emergent properties and surprise. Considering that it is likely that a team of information workers using a suite of visualization and other software tools is some type of complex adaptive system suggests that more holistic approaches to evaluation may be needed.

Already from this brief overview, one can see that useful research advice on the evaluation of information visualization can be gathered from perceptual psychology, cognitive reasoning research, as well as human computer interaction research. Many, but not enough, information visualization researchers are already actively engaged in this pursuit. The purpose of this paper is to applaud them, to encourage more such research, and to suggest that the research community to be more welcoming of a greater variety of these types of research results.

### 3 Choosing an Evaluation Approach

A recent call for papers from the information visualization workshop, Beyond Time and Errors (BELIV06) held at Advanced Visual Interfaces 2006, stated that “*Controlled experiments remain the workhorse of evaluation but there is a growing sense that information visualization systems need new methods of evaluation, from longitudinal field studies, insight based evaluation and other metrics adapted to the perceptual aspects of visualization as well as the exploratory nature of discovery*” [7]. The purpose of this section is to encourage people to consider more broadly what might be the most appropriate research methods for their purposes. To further this purpose a variety of types of empirical research that can be usefully conducted are briefly outlined and these differing types are discussed in terms of their strengths and weaknesses. This discussion draws heavily from McGrath’s paper Methodology Matters [50] that was initially written for social scientists. However, while social scientists work towards understanding humans as individuals, groups, societies and cultures, in information visualization – similarly to HCI – we are looking to learn about how information visualizations do or do not support people in their information tasks and/or how people conduct their information related tasks so that visualization can be better designed to support them. To gain this understanding we sometimes study people using information visualization software and sometimes it may be important to study people independently of that software, to better understand the processes we are trying to support.

There are some commonalities to all studies. They all must start with some question or questions that will benefit from further study. Also, they all must relate their research questions to the realm of existing ideas, theories and findings. These ideas, theories, and concepts are needed to relate the new study to existing research. For example, the results from a new study might be in contrast to existing ideas, in agreement with existing ideas, or offer an extension of or variation to existing ideas. A study must also have a method. This is what this section is about – possible types of empirical methodologies.

All methods offer both advantages and disadvantages. One important part of empirical research is choosing the most appropriate research methods for your content, your ideas, and your situation. The fact that methods both provide and limit evidence suggests that making use of a wide variety of methodologies will, in time, strengthen our understandings. Thus, both conducting a greater variety of studies and encouraging this by publishing research that employs a greater variety of methodologies will help to develop a better understanding of the value of information visualization and its potential in our communities.

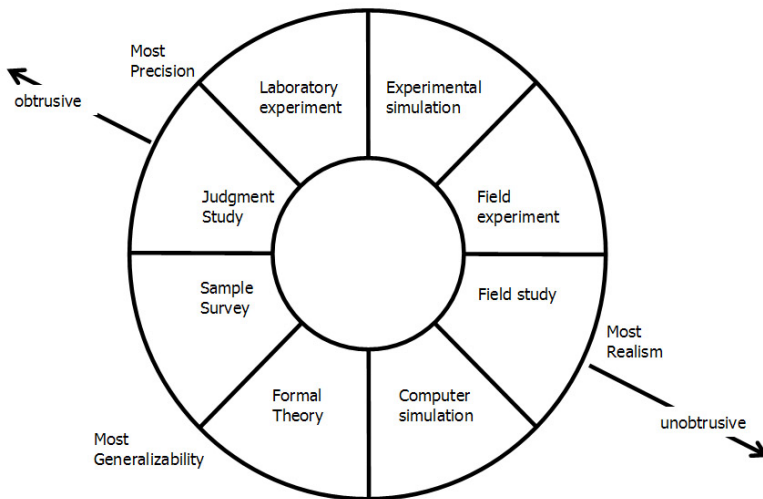
When conducting a study there are three particularly desirable factors: generalizability, precision, and realism [50]. Ideally, one would like all of these factors in one’s results. However, existing methodologies do not support the actualization of all three simultaneously. Each methodology favours one or two of these factors, often at the expense of the others; therefore the choice of a methodology for a particular goal is important. To define these terms (as used in McGrath [50]):

- **Generalizability:** a result is generalizable to the extent to which it can apply to other people (than those directly in the study) and perhaps even extend to other situations.



- **Precision:** a result is precise to the degree to which one can be definite about the measurements that were taken and about the control of the factors that were not intended to be studied.
- **Realism:** a result is considered realistic to the extent to which the context in which it was studied is like the context in which it will be used.

Figure 1 (adapted and simplified from McGrath [50]) shows the span of common methodologies currently in practice in the social sciences. They are positioned around the circle according to the labels: most precision, most generalizability and most realism. The closer a methodology is placed to a particular label, the more that label applies to that methodology. Next, these methodologies are briefly described. For fuller descriptions see McGrath 1995.



**Fig. 1.** Types of methodologies organized to show relationships to precision, generalizability and realism. (adapted, simplified from McGrath 1995)

**Field Study:** A field study is typically conducted in the actual situation, and the observer tries as much as possible to be unobtrusive. That is, the ideal is that the presence of the observer does not affect what is being observed. While one can put considerable effort into minimizing the impact of the presence of an observer, this is not completely possible [50]. Examples of this type of research include ethnographic work in cultural anthropology, field studies in sociology, and case studies in industry. In this type of study the realism is high but the results are not particularly precise and likely not particularly generalizable. These studies typically generate a focused but rich description of the situation being studied.

**Field Experiment:** A field experiment is usually also conducted in a realistic setting; however, an experimenter trades some degree of unobtrusiveness in order to obtain more precision in observations. For instance, the experimenter may ask the participants to perform a specific task while the experimenter is present. While realism is

still high, it has been reduced slightly by experimental manipulation. However, the necessity of long observations may be shortened and results may be more readily interpretable and specific questions are more likely to be answered.

**Laboratory Experiment:** In a laboratory experiment the experimenters fully design the study. They establish what the setting will be, how the study will be conducted, what tasks the participants will do, and thus plan the whole study procedure. Then the experimenter gets people to participate as fully as possible following the rules of the procedure within the set situation. Carefully done, this can provide for considerable precision. In addition, non-realistic behaviour that provides the experimenter more information can be requested such as a ‘think aloud’ protocol [43]. Behaviour can be measured, partly because it is reasonably well known when and where the behaviour of interest may happen. However, realism is largely lost and the degree to which the experimenter introduces aspects of realism will likely reduce the possible precision.

**Experimental Simulation:** With an experimental simulation the experimenter tries to keep as much of the precision as possible while introducing some realism via simulation. There are examples where this approach is essential such as studying driving while using a cell phone or under some substance’s influence by using a driving simulator. Use of simulation can avoid risky or un-ethical situations. Similarly although less dramatically, non-existent computer programs can be studied using the ‘Wizard of Oz’ approach in which a hidden experimenter simulates a computer program. This type of study can provide us with considerable information while reducing the dangers and costs of a more realistic experiment.

**Judgment Study:** In a judgment study the purpose is to gather a person’s response to a set of stimuli in a situation where the setting is made irrelevant. Much attention is paid to creating ‘neutral conditions’. Ideally, the environment would not affect the result. Perceptual studies often use this approach. Examples of this type of research include the series of studies that examine what types of surface textures best support the perception of 3D shape (e.g. [34, 38]), and the earlier related work about the perception of shape from shading [39]. However, in assessing information visualizations this idea of setting a study in neutral conditions must be considered carefully, as witnessed by Reilly and Inkpen’s [62] study which showed that the necessity for an interactive technique developed to support a person’s mental model during transition from viewing one map to another (subway map to surface map) was dependent on the distractions in the setting. This transition technique relates to ideas of morphing and distortion in that aspects of the map remain visible while shifting. These studies in a more neutral experiment setting showed little benefit, while the same tasks in a noisy, distracting setting showed considerable benefit.

**Sample Survey:** In a sample survey the experimenter is interested in discovering relationships between a set of variables in a given population. Examples of these types of questions include: of those people who discover web information visualization tools how many return frequently and are their activities social or work related? Of those people who have information visualization software available at work what is the frequency of use? Considering the increased examples of information visualization results and software on the web, is the general population’s awareness of and/or use of information visualization increasing? In these types of studies proper sampling

of the population can lead to considerable generalizability. However, while choosing the population carefully is extremely important, often it is difficult to control. For example in a web-based survey, all returned answers are from those types of people who are willing to take the time, fill out the questionnaire, etc. This is a type of bias and thus reduces generalizability. Also, responses are hard to calibrate. For instance, a particular paper reviewer may never give high scores and the meta-reviewer may know this and calibrate accordingly or may not know this. Despite these difficulties, much useful information can be gathered this way. We as a community must simply be aware of the caveats involved.

**Formal Theory:** Formal theory is not a separate experimental methodology but an important aspect of all empirical research that can easily be overlooked. As such, it does not involve the gathering of new empirical evidence and as a result is low in both precision and realism. Here, existing empirical evidence is examined to consider the theoretical implications. For example, the results of several studies can be considered as a whole to provide a higher-level or meta-understanding or the results can be considered in light of existing theories to extend, adjust or refute them. Currently this type of research is particularly difficult to publish in that there are no new information visualizations and no new empirical results. Instead, the contribution moves towards the development of theories about the use of and practicality of information visualizations.

**Computer Simulation:** It is also possible to develop a computer simulation that has been designed as logically complete. This method is used in battle simulation, research and rescue simulation, etc. This type of strategy can be used to assess some visualizations. For instance, a visualization of landscape vegetation that includes models of plant growth and models of fire starts and spread can be set to simulate passage of several hundred years. If the resulting vegetation patterns are comparable to existing satellite imagery this provides considerable support for the usefulness of models [22]. Since this type of research strategy does not involve participants, discussion of generalizability over populations is not applicable. Also, since the models are by definition incomplete, notions of precision in measurement are often replaced with stochastic results. On the other hand it does provide a method of validation and offers a parallel with which we can study realistic situations, such as explosions, turbulence in wind tunnels, etc.

## 4 Focus on Quantitative Evaluation

Quantitative evaluations, most well known as laboratory experiments or studies, are those methodologies in which precision is relatively high and in which some declaration can be made about the possible generalization to a larger population. These declarations can include information about the characterization of this larger population and how likely it is that the generalization will hold. These types of experiments or studies are part of the traditional empirical scientific experimental approach and have evolved and been refined through the centuries of scientific research. Science does and has depended on these methods. Slowly, through careful and rigorous application of the experimental process, knowledge has been built up, usually one piece at a time.

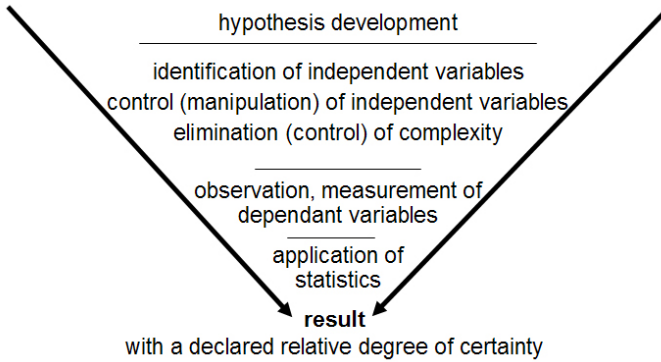
The experiments or studies involve a rigorous process of hypothesis development, identification and control of the independent variables, observation and measurement of the dependent variables, and application of statistics which enable the declaration of the confidence with which the results can be taken. In these formal studies or controlled evaluations, the experimenter controls the environment or setting, manipulates chosen factor(s) or variable(s) – the independent variable(s) - in order to be able to measure and observe the affect this manipulation has on one or more other factors – the dependent variable(s). Ideally no other factors change during the experiment. Once the changes to the dependent variables have been measured, statistical methods can be applied to understand the relative importance of the results. Done with sufficient thoroughness, this process can arrive at facts about which we can be relatively certain. The application of this scientific process will try to reduce the overall complexity by fine tuning particular questions or hypotheses, using these hypotheses to allow one to cull some of the complexity by trying to eliminate as many of the extraneous variables as possible. Traditionally experiments of this type are used to shed light on cause and effect relationships; that is, to discover whether changes in some factor result in changes to another factor.

This idea that we can observe simpler, more manageable subsets of the full complex process is appealing, and it is clear from centuries of experiments that much can be learnt in this manner.

#### 4.1 Quantitative Methodology

Since quantitative empirical evaluations have evolved over the centuries the methodology has become relatively established (Figure 2). This brief overview is included for completeness; the interested reader should refer to the many good books on this subject [15, 17, 33]. This methodology includes:

- **Hypothesis Development:** Much of the success of a study depends on asking an interesting and relevant question. This question should ideally be of interest to the broader research community, and hopefully answering it will lead to a deeper or new understanding of open research questions. Commonly the importance of the study findings results from a well thought through hypothesis, and formulating this question precisely will help the development of the study.
- **Identification of the Independent Variables:** The independent variables are the factors to be studied which may (or may not) affect the hypothesis. Ideally the number of independent variables is kept low to provide more clarity and precision in the results.
- **Control of the Independent Variables:** In designing the experiment the experimenter decides the manner in which the independent variables will be changed.
- **Elimination of Complexity:** In order to be clear that it is actually the change in the independent variable that caused the study's result, it is often the case that other factors in the environment need to be controlled.
- **Measurement of the Dependent Variables:** Observations and measurements are focused on the dependent variables as they change or do not change in



**Fig. 2.** A simple schematic of the traditional experimental process.

response to the manipulation of the independent variable. The aspects to be measured are often called metrics. Common metrics include: speed, accuracy, error rate, satisfaction, etc.

- **Application of Statistics:** The results collected can then be analysed through the application of appropriate statistics. It is important to remember that statistics tell us how sure we can be that these results could (or could not) have happened by chance. This gives a result with a relative degree of certainty. There are many good references such as Huck [33].

These steps sound deceptively simple but doing them well requires careful and rigorous work. For instance, it is important that the study participants are valued, that they are not over-stressed, and that they are given appropriate breaks, etc. Also, exactly what they are being asked to do must be clear and consistent across all participants in your study. Since small inconsistencies such as changes in the order of the instructions can affect the results, the common recommendation is that one scripts the explanations. Perhaps most importantly, to eliminate surprises and work out the details, it is best to pilot – run through the experiment in full – repeatedly.

## 4.2 Quantitative Challenges

Even though these types of experiments have been long and effectively used across all branches of science, there remain many challenges to conducting a useful study. We mention different types of commonly-discussed errors and validity concerns and relate these to the McGrath's discussion as outlined in Section 3. In this discussion we will use a simple, abstract example of an experiment that looks at the effect of two visualization techniques, VisA and VisB, on performance in search. There are several widely discussed issues that can interfere with the validity of a study.

**Conclusion Validity: Is there a relationship?** This concept asks whether within the study there is a relationship between the independent and the dependent variables. Important factors in conclusion validity are finding a relationship when one does not exist (type I error) and not finding a relationship when one does exist (type II error).

**Table 1.** Type I and Type II Errors

		Reality	
		H <sub>0</sub> TRUE	H <sub>0</sub> FALSE
Experimental decision	H <sub>0</sub> TRUE	ok	Type II
	H <sub>0</sub> FALSE	Type I	ok

**Type I and Type II Errors:** If one is interested in which visualization technique VisA or VisB helps people conduct a particular task faster one might formulate a null hypothesis (H<sub>0</sub>) – *there is no difference in speed of search between VisA and VisB*. The possible type I, false negative, and type II, false positive, errors are specified in Table 1. The columns represent whether the null hypothesis is true or false in reality and the rows show the decision made based on the results of the experiment. Ideally the results of the experiment reflect reality and that if the hypothesis is false (or true) in reality it will show as false (or true) in the experiment. However, it is possible that the hypothesis is true in reality – VisA does support faster search than VisB – but that this fails to be revealed by the experiment. This is a type II error. A type I error occurs if the null hypothesis is true in reality (there is no difference) and one concludes that there *is* a difference. Type I errors are considered more serious. That is, it is considered worse to claim that VisA improves search when it does not, than to say there was no measurable difference.

**Internal Validity: Is the relationship causal?** This concept is important when an experiment is intended to reveal something about causal relationships. Thus, internal validity will be important in our simple example because the study is looking at what effect VisA and VisB have on search. The key issue here is whether the results of one's study can properly be attributed to what happened within the experiment. That is, that no other factors influenced or contributed to the results seen in the study. Another way of asking this question is: are there possible alternate causes for the results seen in the study?

**Construct Validity: Can we generalize to the constructs (ideas) the study is based on?** This concept considers whether the experiment has been designed and run in a manner that answers the intended questions. This is an issue about whether the right factors are being measured or whether the factors the experimenter intends to measure are actually those being measured. For instance, does the experiment measure the difference due to the techniques VisA and VisB or the difference in participant's familiarity with VisA and VisB. For instance, if the construct is that a person will have higher satisfaction when using VisB, does measuring error rates and completion times provide answers for this construct? An important part of this concept of construct validity is **measurement validity**. Measurement validity is concerned with questions such as is one measuring what one intends to measure and is the method of measurement reliable and consistent. That is, will the same measurement process provide the same results when repeated?

**External Validity: Can we generalize the study results to other people/places/times?** External validity is concerned with the extent to which the result of a study can be generalized. If a study has good internal and construct validity the results apply to the setting, time, and participants being studied. The extent to which the results apply beyond the immediate setting, time and participants depends, for participants, on the participant sample and the population from which it was drawn. For instance, in practice it is common to draw participants from the geographic region in which the study is run. Does this mean that the results only apply to people from that region? If culture has a possible impact on the results, they may not generalize. If one addresses the need to include cultural variation by recruiting participants from different cultures from a university's foreign students, one might have at least partially addressed the need to run the study across cultural variations but now have limited the demographic to university students which may introduce its own skew. Understanding the population to which one would like to be able to generalize the study results and successfully obtaining an appropriate participant sample is a difficult issue. This does not mean we can not learn from more specific participant samples. It does mean that reporting the demographics of the sample and being cautious about generalizations is important. Participant sample choice is just one factor influencing external validity. Setting includes other factors such as noise, interruption, and distractions. Possible temporal factors include events that occurred before or are anticipated after the experiment.

**Ecological Validity:** Ecological validity discussions focus on the degree to which the experimental situation reflects the type of environment in which the results will be applied. This concept relates strongly to McGrath's concept of realism. It is distinct from the idea of external validity, in that external validity is concerned with whether the experimental results generalize to other situations, while ecological validity is concerned with how closely the experimental settings matches the real setting in which the results might be applied. Thus it is possible to have good ecological validity; the study is conducted on site, but that the results are applicable only to that site. This would indicate poor external validity in that the results do not generalize beyond the specific setting.

### 4.3 Quantitative Studies Summary Remarks

The number of quantitative studies in information visualization is increasing. Early examples include the series of studies done by Purchase and her collaborators that examine the impact of graph drawing aesthetics on comprehension and usability [58, 59, 60, 61]. Dumais et al. [16] explored use of context techniques in web search. Forlines et al. [23] looked at the effect of display configuration on relationship between visual search and information visualization tasks. Recently, Willet et al. [81] studied embedding information visualizations in widgets.

Quantitative experiments have formed the backbone of experimental science and it is to be expected that they will continue to do so. However, it is relatively easy to find fault in any given experiment because all factors can not usually be completely controlled. If they are completely controlled, external and ecological validity can be impacted. This is particularly true for studies involving humans. Designing and working with experiments is often a matter of making choices about what factors are important and understanding the strengths and limitations of any given study and its results. As

a community it is important that we recognise that we are working towards a larger understanding and that any given study will not present the bigger answer. It instead will contribute to a gradual building of a bigger understanding. For this bigger understanding we need to encourage authors to openly discuss the limitations of their studies, because both the results and the limitations are important. This is also true for negative results. It can be just as important to understand when there are no differences among techniques and when these differences exist.

## 5 Focus on Qualitative Evaluation

Qualitative inquiry works toward achieving a richer understanding by using a more holistic approach that considers the interplay among factors that influence visualizations, their development, and their use [56]. Qualitative techniques lend themselves to being more grounded in more realistic settings and can also be incorporated into all types of studies. This includes qualitative studies conducted as part of the design process [64, 73], in situ interviews [83], field studies [72], and use of observational studies to create design and evaluative criteria that are derived from observed data [71]. These types of studies offer potential for improved understanding of existing practices, analysis environments, and cognitive task constraints as they occur in real or realistic settings. In providing a brief overview of a variety of qualitative methods, we hope to spark further research and application of qualitative methods in information visualization; to expand our empirical approaches to include the application of qualitative methods to design and evaluation; and to encourage a wider acceptance of these types of research methodologies in our field.

### 5.1 Qualitative Methods

At the heart of qualitative methods is the skill and sensitivity with which data is gathered. Whether the records of the data gathered are collected as field notes, artefacts, video tapes, audio tapes, computer records and logs, or all of these, in qualitative empirical approaches there are really only two primary methods for gathering data: observations and interviews. Observation and interview records are usually kept continually as they occur, as field notes, as regular journal entries as well as often being recorded as video or audio tapes. Artefacts are collected when appropriate. These can be documents, drawings, sketches, diagrams, and other objects of use in the process being observed. These artefacts are sometimes annotated as part of use practices or in explanation. Also, since the communities we are observing are often technology users, technology-based records can also include logs, traces, screen captures, etc. Both observation and interviewing are skills and as such develop with practice and can, at least to some extent, be learnt. For full discussions on these skills there are many useful books such as Seidman [65] and Lofland and Lofland [45].

#### 5.1.1 Observation Techniques

The following basic factors have been phrased in terms of developing observational records but implicitly also offer advice on what to observe:



- Try to keep jotting down notes unobtrusively. Ideally, notes are taken as observations occur; however, if one becomes aware that one's note taking is having an impact on the observations, consider writing notes during breaks, when shielded, or at the end of the day.
- Minimize the time gap from observations to note taking. Memory can be quite good for a few hours but does tend to drop off rapidly.
- Include in observations the setting, a description of the physical setup, the time, who is present, etc. Drawing maps of layouts and activities can be very useful.
- Remember to include both the overt and covert in activities and communications. For example, that which is communicated in body language and gestures, especially if it gets understood and acted upon, is just as important as spoken communications. But be careful of that grey area where one is not sure to what extent a communication occurred.
- Remember to include both the positive and negative. Observed frustrations and difficulties can be extremely important in developing a fuller understanding.
- Do not write notes on both sides of a paper. This may seem trivial but experienced observers say this is a must [6]. You can search for hours, passing over many times that important note that is on the back of another note.
- Be concrete whenever possible.
- Distinguish between word-for-word or verbatim accounts and those you have paraphrased and/or remembered.

### 5.1.2 Interview Techniques

These are a few brief points of advice about interviewing. Do remember that while sorting out the right questions to ask is important, actively listening to what the participant says is the most important of all interviewing skills.

- Make sure that you understand what they are telling you and that the descriptions, explanations they are giving you are complete enough. However, when asking for clarification, try to avoid implying that their explanations are poor because one does not want to make one's participants defensive. Ask instead for what they meant by particular word usage or if they would explain again. The use of the word *again* implies that the interviewer did not catch it all rather than the explanation was incomplete.
- Limit your inclination to talk. Allow for pauses in the conversation, sometimes note taking can be useful here. The participant will expect you to be taking notes. In this situation note taking can actually express respect for what the participant has said.
- Remember that the default is that the participant will regard the interview to some extent as public and thus will tell you the public version. Do listen for and encourage the less formal, less guarded expression of their thoughts. One example, from Seidman [65], is the use of the word 'challenge'. Challenge is an expected term for a problem. The details of the problem might be explained more fully if one asks what is meant in the given situation by the word challenge.

- Follow up on what the participant says. Do allow the interview to be shaped by the information your participant is sharing.
- Avoid leading questions. An important part of minimizing experimenter bias is wording questions carefully so as to avoid implying any types of answers. For example, asking a participant what a given experience was like for them, leaves space for their personal explanations.
- Ask open ended questions. This can involve asking for a temporal reconstruction of an event or perhaps a working a day or asking for a subjective interpretation of an event.
- Ask for concrete details. These can help trigger memories.
- With all the above do remember that one of the most important pluses of an interview process is the humanity of interviewer. Being present, aware and sensitive to the process is your biggest asset. These guidelines are just that; guidelines to be used when useful and ignored when not.

## 5.2 Types of Qualitative Methodologies

This section is not intended to be a complete collection of all types of qualitative inquiry. Rather it is meant to give an overview of some of the variations possible, set in a discussion about when and where they have proven useful. This overview is divided into three sections. First, the type of qualitative methodologies often used in conjunction with or as part of more quantitative methodologies is discussed. Then, we mention the approaches taken in the area of heuristic, or, as they are sometimes referred to ‘discount’, inspection methodologies. The last section will cover some study methodologies that are intentionally primarily qualitative.

### 5.2.1 Nested Qualitative Methods

While qualitative methodologies can be at the core of some types of studies, some aspects of qualitative inquiry are used in most studies. For instance, data gathered by asking participants for their opinions or preferences is qualitative. Gorard [26] argues that quantitative methods can not ignore the qualitative factors of the social context of the study and that these factors are, of necessity, involved in developing an interpretation of the study results. There are many methods used as part of studies such as laboratory experiments that provide us with qualitative data. The following are simply a few examples to illustrate how common this mixed approach is.

**Experimenter Observations:** An important part of most studies is that the experimenter keeps notes of what they observe as it is happening. The observations themselves can help add some degree of realism to the data and the practice of logging these observations as they happen during the study helps make them more reliable than mere memory. However, they are experimenter observations and as such are naturally subjective. They do record occurrences that were not expected or are not measurable so that they will also form part of the experimental record. These observations can be helpful during interpretation of the results in that they may offer explanations for outliers, point towards important experimental re-design, and suggest future directions for study. Here, experimenter observations augment and enrich the primar-

ily quantitative results of a laboratory experiment and in this they play an important but secondary role.

**Think-Aloud Protocol:** This technique, which involves encouraging participants to speak their thoughts as they progress through the experiment, was introduced to the human-computer-interaction community by [43]. Discussions about this protocol in psychology date back to 1980 [19, 20, 21]. Like most methodologies, this one also involves tradeoffs. While it gives the experimenter/observer the possibility of being aware of the participants' thoughts, it is not natural for most people and can make a participant feel awkward; thus, think aloud provides additional insight while also reducing the realism of the study. However, the advantage for hearing about a participant's thoughts, plans, and frustrations frequently out-weigh the disadvantages and this is a commonly used technique. Several variations have been introduced such as 'talk aloud' which asks a participant to more simply announce their actions rather than their thoughts [21].

**Collecting Participant Opinions:** Most laboratory experiments include some method by which participant opinions and preferences are collected. This may take the form of a simple questionnaire or perhaps semi-structured interviews. Most largely quantitative studies such as laboratory experiments do ask these types of questions, often partially quantifying the participant's response by such methods as using a Likert scale [44]. A Likert scale asks a participant to rate their attitude according to degree. For instance, instead of simply asking a participant, 'did you like it?' A Likert scale might ask the participant to choose one of a range of answers 'strongly disliked,' 'disliked,' 'neutral,' 'liked,' or 'strongly liked.'

**Summary of Nested Qualitative Methods:** The nested qualitative methods mentioned in this section may be commonplace to many readers. The point to be made here is that in the small, that is as part of a laboratory experiment, inclusion of some qualitative methods is not only commonplace, its value is well recognized. This type of inclusion of qualitative approaches adds insight, explanations and new questions. It also can help confirm results. For instance, if participants' opinions are in line with quantitative measures – such as the fastest techniques being the most liked – this confirms the interpretation of the fastest technique being the right one to chose. However, if they contradict – such as the fastest techniques not being preferred – interesting questions are raised including questioning the notion that fastest is always best.

### 5.2.2 Inspection Evaluation Methods

We include a discussion of inspection methods because, while they are not studies per se, they are useful, readily available, and relatively inexpensive evaluation approaches. The common approach is to use a set of heuristics as a method of focusing attention on important aspects of the software – interface or visualization – which need to be considered [54]. These heuristics or guidelines can be developed by experts or from the writings of experts. Ideally, such an inspection would be conducted by individual experts or even a group of experts. However, it has been shown that in practice, that a good set of heuristics can still be effective in application if a few, such as three or four, different people apply them [54]. For information visualization it is important to consider exactly what visualization aspects a given set of heuristics will shed light on.

**Usability Heuristics:** These heuristics, as introduced and developed by Nielson and Mack [1994], focus on the usability of the interface and are designed to be applied to any application, thus are obviously of use to information visualizations. They will help make sure that general usability issues are considered. These heuristics are distilled down to ten items – visibility of system status, match between system and real world, personal control and freedom, consistency and standards, error prevention, recognition rather than recall, flexibility and efficiency, aesthetic and minimalist design, errors handling, and help and documentation.

**Collaboration Heuristics:** When interfaces are designed for collaboration, two additional major categories arise in importance: communication and coordination. Baker et al. [4] developed a set of heuristics that explore these issues based on the Mechanics of Collaboration [29]. As information visualizations start to be designed for collaborative purposes, both distributed [31, 78] and co-located [35], these heuristics will also be important.

**Information Visualization Heuristics:** While the usability heuristics apply to all infovis software and the collaboration heuristics apply to the growing body of collaborative information visualizations, there are areas of an information visualization that these at best gloss over. In response, the Information Visualization research community has proposed a variety of specific heuristics. Some pertain to given data domains such as ambient displays [46] and multiple view visualizations [5]. Others focus on a specific cognitive level, for instance knowledge and task [1], or task and usability [66]. Tory and Möller [74] propose the use of heuristics based on both visualization guidelines and usability. As explored by Zuk and Carpendale [84], we can also consider developing heuristics based on the advice from respected experts such as design advice collected from Tufte’s writings [75, 76, 77], semiotic considerations as expressed by Bertin [8] and/or research in cognitive and perceptual science as collected by Ware [79]. Alternatively, we can start from information visualization basics such as presentation, representation and interaction [68]. However, a concept such as presentation cuts across design and perception, while representation advice, such as what types of visuals might best represent what types of data, might be distilled from the guidelines put forth by Bertin [8] and from an increasing body of cognitive science as gathered in Ware [79]. Sorting out how to best condense these is a task in itself [52, 85]. “At this stage of development of heuristics for information visualization we have reached a similar problem as described by Nielson and Mack [54]. It is a difficult problem to assess which list(s) are better for what reasons and under what conditions. This leads to the challenges of developing an optimal list that comprises the most important or common Information Visualization problems” (page 55, [85]).

**Summary of Inspection Evaluation Methods:** While experience in the human computer interaction communities and the growing body of information visualization specific research indicates that heuristics may prove a valuable tool for improving the quality of information visualizations, there is considerable research yet to be conducted in the development of appropriate taxonomies and application processes for heuristics in information visualization.

The currently recommended application approach for usability heuristics is that evaluators apply the heuristics in a two pass method. The first pass is done to gain an overview and second is used to assess in more detail each interface component with

each heuristic [54]. The original use indicated that in most situations three evaluators would be cost effective and find most usability problems [54]. However, subsequent use of heuristics for web site analysis appears to sometimes need more evaluators [9, 69]. Further, this may depend on the product. While application of heuristics has not yet been formally studied in terms of web sites, it does introduce the possibility that information visualization heuristics may also need to be data, task or purpose specific.

Heuristics are akin to the design term *guidelines* in that both provide a list of advice. Design guidelines are often usefully applied in a relatively ad hoc manner as factors to keep in mind during the design process and heuristic lists can definitely be similarly used. While there are definitely benefits that accrue in the use of guidelines and heuristics, it is important to bear in mind that they are based on what is known to be successful and thus tend not to favour the unusual and the inventive. In the design world, common advice is that while working without knowledge of guidelines is foolish, following them completely is even worse.

### 5.2.3 Qualitative Methods as Primary

A common reason for using qualitative inquiry is to develop a richer understanding of a situation by using a more holistic approach. Commonly, the qualitative research method's goal is to collect data that enables full, rich descriptions rather than to make statistical inferences [3, 14]. There are a wealth of qualitative research methods that can help us to gain a better understanding of the factors that influence information visualization use and design. Just as we have pointed out how qualitative methods can be effectively used within quantitative research, qualitative research can also include some quantitative results. For instance, there may be factors that can be numerically recorded. These factors can then be presented in combination with qualitative data. For example, if a questionnaire includes both fixed-choice questions and open ended questions, quantitative measurement and qualitative inquiry are being combined [56].

Qualitative methods can be used at any time in the development life cycle. A finished or near to finished product can be assessed via case studies or field studies. Also, there is a growing use of these methods as a preliminary step in the design process. The HCI and particularly the computer-supported cooperative work (CSCW) research communities have successfully been using qualitative methods to gain insight that can inform the initial design. CSCW researchers have learned a lot about how to support people working together with technology through pre-design observation and qualitative analysis of how people work together without technology. The basic idea is that through observations of participants' interactions with physical artefacts, a richer understanding of basic activities can be gained and that this understanding can be used to inform interface design. This approach generally relies on observation of people, inductive derivation of hypotheses via iterative data collection, analysis, and provisional verification [14]. For example, Tang's study of group design activities around shared workspaces revealed the importance of gestures and the workspace itself in mediating and coordinating collaborative work [73]. Similarly, Scott et al. [64] studied traditional tabletop gameplay and collaborative design, specifically focusing on the use of tabletop space, and the sharing of items on the table. Both studies are an example of how early evaluation can inform the design of digital systems. In both cases, the authors studied traditional, physical contexts first, to understand participants' interactions with the workspace, the items in the workspace, and

within the group. The results of these experiments are regarded as providing important information about what group processes to support and some indication about how this might be done. This type of research can be particularly important in complex or sensitive scenarios such as health care situations [72]. Brereton and McGarry [11] observed groups of engineering students and professional designers using physical objects to prototype designs. They found that the interpretation and use of physical objects depended greatly on the context of its placement, indicating that the context of people's work is important and is difficult to capture quantitatively. Their goal was to determine implications for the design of tangible interfaces. Other examples include Saraiya et al. [63] who used domain expert assessments of insight to evaluate bioinformatics visualizations, while Mazza and Berre [48] used focus groups and semi-structured interviews in their analysis of visualization approaches to support instructors in web-based distance education.

The following are simply examples of empirical methods in which gathering of qualitative data is primary. There are many others; for instance, Moggridge [51] mentions that his group makes active use of fifty-one qualitative methods in their design processes.

**In Situ Observational Studies:** These studies are at the heart of field studies. Here, the experimenter gets permission to observe activities as they take place in situ. In these studies the observer does their best to remain unobtrusive during the observations. The ideal in Moggridge's terms is to become as a 'fly on the wall' that no one notices [51]. This can be hard to achieve in an actual setting. However, over time a good observer does usually fade into the background. Sometimes observations can be collected via video and audio tapes to avoid the more obvious presence of a person as observer but sometimes making such recordings is not appropriate as in medical situations. In these studies the intention is usually to gather a rich description of the situation being observed. However, there is both a difference and an overlap in the type of observations to be gathered when the intention is (a) to better understand the particular activities in a given of setting, or (b) to use these observations to inform technology design. Thus, because different details are of prime interest it is important that our research community conducts these types of observational studies to better inform initial design as well as to better understand the effectiveness of new technology in use. These studies have high realism, result in rich context explicit data and are time and labour intensive when it comes to both data collection and data analysis.

**Participatory Observation:** This practice is the opposite of participatory design. Here an information visualization expert becomes part of the application expert's team to experience the work practices first hand rather than application experts becoming part of the information visualization design team. In participatory observation, additional insights can be gained through first-hand observer experience of the tasks and processes of interest in the context of the real world situation. Here, rather than endeavouring to be unobtrusive, the observer works towards becoming an accepted part of the community. Participatory observation is demonstrably an effective approach since as trust and rapport develop, an increasingly in-depth understanding is possible. Our research community is interested in being able to better understand the work practices of many different types of knowledge workers. These workers are usually highly trained, highly paid, and often under considerable time pressures. Not

surprisingly, they are seldom willing to accept an untrained observer as part of their team. Since information visualization researchers are of necessity highly trained themselves, it is rare that an information visualization researcher will have the necessary additional training to become accepted as a participatory observer. However, domain expertise is not always essential for successful participatory observation. Expert study participants can train an observer on typical data analysis tasks – a process which may take several hours, and then “put them to work” on data analysis using their existing tools and techniques. The observer keeps a journal of the experience and the outcomes of the analysis were reviewed with the domain experts for validity. Even as a peripheral participant, valuable understandings of domain, tasks, and work culture can be developed which help clarify values and assumptions about data, visualizations, decision making and data insights important to the application domain. These understandings and constructs can be important to the information visualization community in the development of realistic tools.

**Laboratory Observational Studies:** These studies use observational methodologies in a laboratory setting. A disadvantage of in situ observations is that they often require lengthy observations. For instance, if the observer is interested in how an analyst uses visual data, they will have to wait patiently until the analyst does this task. Since an analyst may have many other tasks – meetings, conference calls, reports, etc. – this may take hours or even days. One alternative to the lengthy in situation wait is to design an observational experiment in which, similarly to a laboratory experiment, the experimenter designs a setting, a procedure and perhaps even a set of tasks. Consider, for example, developing information visualizations to support co-located collaboration. Some design advice on co-located collaborative aspects is available in the computer supported cooperative work literature [35]. However, while this advice is useful, it does not inform us specifically about how teams engage in collaborative tasks when using visual information. Details such as how and when visualizations will be shared and what types of analysis processes need to be specifically supported in collaborative information visualization systems were missing. Here, an observational approach is appropriate because the purpose is to better understand the flow and nature of the collaboration among participants, rather than answering quantifiable lower-level questions. In order to avoid temporal biases in existing software, pencil and paper based visualizations were used. This allowed for the observation of free arrangement of data, annotation practices, and collaborative processes unconstrained by any particular visualization software [36].

**Contextual Interviews:** As noted in Section 5.1, interviewing in itself is core to qualitative research. Conducting an interview about a task, setting, or application of interest within the context in which this work usually takes place is just one method that can enrich the interview process. Here the realism of the setting helps provide the context that can bring to mind the day-to-day realities during the interview process (for further discussion see Holtzblatt and Beyer 1998). For example, to study how best to support the challenging problem of medical diagnosis, observing and interviewing physicians in their current work environment might help to provide insights into their thought processes that would be difficult to capture with other methodologies. A major benefit of qualitative study can be seeing the big picture – the context in which a new visualization support may be used. The participants' motives, misgiv-

ings, and opinions shed light on how they relate to existing support, and can effectively guide the development of new support. This type of knowledge can be very important at the early stage of determining what types of information visualizations may be of value.

**Summary of qualitative methods as primary:** These four methods are just examples of a huge variety of possibilities. Other methods include action research [42], focus groups [48], and many more. All these types of qualitative methods have the potential to lessen the task and data comprehension divide between ourselves as visualization experts and the domain experts for whom we are creating visualizations. That is, while we can not become analysts, doctors, or linguists, we can gain a deeper understanding of how they work and think. These methods can open up the design space, revealing new possibilities for information visualizations, as well as additional criteria on which to measure success.

### 5.3 Challenges for Qualitative Methods

A considerable challenge to qualitative methods is that they are particularly labour intensive. Gathering data is a slow process and rich note taking is an intensive undertaking, as are transcribing and subsequent analysis.

#### 5.3.1 Sample Sizes

Sample sizes for qualitative research are determined differently than for quantitative research. Since qualitative research is not concerned with making statistically significant statements about a phenomenon, the sample sizes are often lower than required for quantitative research. Often, sample sizes are determined during the study. For instance, a qualitative inquiry may be continued until one no longer appears to be gaining new data through observation [3]. There is no guideline to say when this ‘saturation’ may occur [70]. Sample sizes may vary greatly depending on the scope of the research problem but also the experience of the investigator. An experienced investigator may reach a theoretical saturation earlier than a novice investigator. Also, because each interview and/or observation can result in a large amount of data, sometimes compromises in sample size have to be made due to considerations about the amount of data that can be effectively processed.

#### 5.3.2 Subjectivity

Experimenter subjectivity can be seen as an asset because of the sensitivity that can be brought to the observation process. The quality of the data gathering and analysis is dependent on the experience of the investigator [56]. However, the process of gathering any data must be concerned with obtaining representative data. The questions circle about whether the observer has heard or understood fully and whether these observations are reported accurately. Considerations include:

- Is this a first person direct report? Otherwise normal common sense about 2<sup>nd</sup>, 3<sup>rd</sup>, and 4<sup>th</sup> hand reports needs to be considered.
- Does the spatial location of the observer provide an adequate vantage point from which to observe, or might it have led to omissions?



- Are the social relationships of the observer free from associations that might induce bias?
- Does the report appear to be self-serving? Does it benefit the experimenter to the extent that it should be questioned?
- Is the report internally consistent? Do the facts within the report support each other?
- Is the report externally consistent? Do the facts in the report agree with other independent reports?

As a result it is important to be explicit about data collection methods, the position of the researcher with respect to the subject matter, analysis processes, and codes. These details make it possible for other researchers to verify results.

In qualitative research it is acknowledged that the researcher's views, research context, and interpretations are an essential part of the qualitative research method as long as they are grounded in the collected data [3]. This does not, however, mean that qualitative evaluations are less trustworthy compared to quantitative research. Auerbach suggests using the concept of 'transferability' rather than 'generalizability' when thinking about the concepts of reliability and validity in qualitative research [3]. It is more important that the theoretical understanding we have gained can also be found in other research situations or systems and can be extended and developed further when applied to other scenarios. This stands in contrast to the concept of generalizability in quantitative research that wants to prove statistically that the results are universally applicable within the population under study.

Sometimes the point has been raised that if results do not generalize how can they be of use when designing software for general use. For example, qualitative methods might be used to obtain a rich description of a particular situation perhaps only observing the processes of two or three people. The results of a study like this may or may not generalize and the study itself provides no proof that they do. What we have is existence proof: that such processes are in use in at least two or three instances. Consider the worst case; that is that this rich description is an outlier that occurs only rarely. For design purposes, outliers are also important and sensitive design for outliers has been often shown to create better designs for all. For example, motion sensors to open doors may have been designed for wheelchairs but actually are useful features for all.

### 5.3.3 Analyzing Qualitative Data

Qualitative data may be analyzed using qualitative, quantitative, or a combination of both methods. Mixed methods research includes a qualitative phase and a quantitative phase in the overall research study in order to triangulate results from different methods, to complement results from one method with another, or to increase the breadth and range of inquiry by using different methods [28].

Many of the qualitative analysis methods can be grouped as types of thematic analysis, in which analysis starts from observations, then themes are sensed through review of the data, and finally coded [10]. Coding is the process of subdividing and labeling raw data, then reintegrating collected codes to form a theory [70]. Moving from the raw data into themes and a code set may proceed using one of three ap-

proaches: data-driven, motivated from previous research, or theory-driven, each with respectively decreasing levels of sensitivity to the data [10]. In the first style, data-driven, commonly called open coding [14]; themes and a code set are derived directly from the data and nothing else. If the analysis is motivated by previous research, the questions and perhaps codes from the earlier research can be applied to the new data to verify, extend or contrast the previous results. With theory-driven coding one may think using a given theory, such as grounded theory [13], or ethno-methodology [24], as a lens through which to view the data.

In either case the coded data may then be interpreted in more generalized terms. Qualitatively coded data may then be used with quantitative or statistical measures to try and distinguish themes or sampling groups.

## 5.4 Qualitative Summary

Qualitative studies can be a powerful methodology by which one can capture salient aspects of a problem that may provide useful design and evaluation criteria. Quantitative evaluation is naturally precision-oriented, but a shift from high precision to high fidelity may be made with the addition of qualitative evaluations. In particular, while qualitative evaluations can be used throughout the entire development life cycle in other research areas such as CSCW [41, 52, 64, 73], observational studies have been found to be especially useful for informing design. Yet these techniques are under-used and under-reported in the information visualization literature. Broader approaches to evaluation, different units of analysis and sensitivity to context are important when complex issues such as insight, discovery, confidence and collaboration need to be assessed. In more general terms, we would like to draw attention to qualitative research approaches which may help to address difficult types of evaluation questions. As noted by Isenberg et al. [36], a sign in Albert Einstein's office which read, *'Everything that can be counted does not necessarily count; everything that counts cannot necessarily be counted'* is particularly salient to this discussion in reminding us to include empirical research about important data that can not necessarily be counted.

## 6 Conclusions

In this paper we have made a two-pronged call: one for more evaluations in general and one for a broader appreciation of the variety of and importance of many different types of empirical methodologies. To achieve this, we as a research community need to both conduct more empirical research and to be more welcoming of this research in our publication venues. As noted in Section 4, even empirical laboratory experiments, as our most known type of empirical methodology, are often difficult to publish. One factor in this is that no empirical method is perfect. That is, there is always a trade-off between generalizability, precision, and realism. An inexperienced reviewer may recommend rejection based on the fact that one of these factors is not present, while realistically at least one will always be compromised. Empirical research is a slow, labour-intensive process in which understanding and insight can develop through time. That said, there are several important factors to consider when publishing empirical research. These include:

- That the empirical methodology was sensitively chosen. The methodology should be a good fit to the research question, the situation and the research goals.
- That the study was conducted with appropriate rigor. All methodologies have their own requirements for rigor and these should be followed. However, while trying to fit the rigor from one methodology onto another is not appropriate, developing hybrid methodologies that better fit a given research situation and benefit from two or more methodologies should be encouraged.
- That sufficient details are published so that the reader can fully understand the processes and if appropriate, reproduce them.
- That the claims should be made appropriately according to the strengths of the chosen methodology. For instance, if a given methodology does not generalize well, then generalizations should not be drawn from the results.

While there is growing recognition in our research community that evaluation information visualization is difficult [55, 57, 67], the recognition of this difficulty has not in itself provided immediate answers of how to approach this problem. Two positive recent trends of note are: one, that more evaluative papers in the form of usability studies have been published [25, 40, 47, 63, 80, 82], and two, that there are several papers that have made a call for more qualitative evaluations and complementary qualitative and quantitative approaches [18, 36, 48, 74].

This paper is intended merely as a pointer to a greater variety of empirical methodologies and encouragement towards their appreciation and even better their active use. There are many more such techniques and these types of techniques are being developed and improved continuously. There are good benefits to be had through active borrowing from ethnographic and sociological research methods, and applying them to our information visualization needs. In this paper we have argued for an increased awareness of empirical research. We have discussed the relationship of empirical research to information visualization and have made a call for a more sensitive application of this type of research [27]. In particular, we encourage thoughtful application of a greater variety of evaluative research methodologies in information visualization.

**Acknowledgments.** The ideas presented in this paper have evolved out of many discussions with many people. In particular this includes: Christopher Collins, Marian Dörk, Saul Greenberg, Carl Gutwin, Mark S. Hancock, Uta Hinrichs, Petra Isenberg, Stacey Scott, Amy Volda, and Torre Zuk.

## References

1. Amar, R.A., Stasko, J.T.: Knowledge Precepts for Design and Evaluation of Information Visualizations. *IEEE Transactions on Visualization and Computer Graphics* 11(4), 432–442 (2005)
2. Andrews, K.: Evaluating Information Visualisations. In: *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization*, pp. 1–5 (2006)
3. Auerbach, C.: *Qualitative Data: An Introduction to Coding and Analysis*. University Press, New York (2003)

4. Baker, K., Greenberg, S., Gutwin, C.: Empirical Development of a Heuristic Evaluation Methodology for Shared Workspace Groupware. In: Proceedings of the ACM Conference on Computer Supported Cooperative Work, pp. 96–105. ACM Press, New York (2002)
5. Baldonado, M., Woodruff, A., Kuchinsky, A.: Guidelines for Using Multiple Views in Information Visualization. In: Proceedings of the Conference on Advanced Visual Interfaces (AVI), pp. 110–119. ACM Press, New York (2000)
6. Barzun, J., Graff, H.: *The Modern Researcher*, 3rd edn. Harcourt Brace Jovanvich, New York (1977)
7. BELIV 2006, accessed <http://www.dis.uniroma1.it/~beliv06/> (February 4, 2008)
8. Bertin, J.: *Semiology of Graphics* (Translation: William J. Berg). University of Wisconsin Press (1983)
9. Bevan, N., Barnum, C., Cockton, G., Nielsen, J., Spool, J., Wixon, W.: The “Magic Number 5”: Is It Enough for Web Testing? In: CHI Extended Abstracts, pp. 698–699. ACM Press, New York (2003)
10. Boyatzis, R.: *Transforming Qualitative Information: Thematic Analysis and Code Development*. Sage Publications, London (1998)
11. Brereton, M., McGarry, B.: An Observational Study of How Objects Support Engineering Design Thinking and Communication: Implications for the Design of Tangible Media. In: Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI’00), pp. 217–224. ACM Press, New York (2000)
12. Chen, C., Czerwinski, M.: Introduction to the Special Issue on Empirical Evaluation of Information Visualizations. *International Journal of Human-Computer Studies* 53(5), 631–635 (2000)
13. Corbin, J., Strauss, A.: *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, 3rd edn. Sage Publications, Los Angeles (2008)
14. Creswell, J.: *Qualitative Inquiry and Research Design: Choosing Among Five Traditions*. Sage Publications, London (1998)
15. Dix, A., Finlay, J., Abowd, G., Beale, R.: *Human Computer Interaction*, 2nd edn. Prentice-Hall, Englewood Cliffs (1998)
16. Dumais, S., Cutrell, E., Chen, H.: Optimizing Search by Showing Results In Context. In: Proc. CHI’01, pp. 277–284. ACM Press, New York (2001)
17. Eberts, R.E.: *User Interface Design*. Prentice-Hall, Englewood Cliffs (1994)
18. Ellis, E., Dix, A.: An Explorative Analysis of User Evaluation Studies in Information Visualization. In: Proceedings of the Workshop on Beyond Time and Errors: Novel Evaluation Methods for Information Visualization, BELIV (2006)
19. Ericsson, K., Simon, H.: Verbal Reports as Data. *Psychological Review* 87(3), 215–251 (1980)
20. Ericsson, K., Simon, H.: Verbal Reports on Thinking. In: Faerch, C., Kasper, G. (eds.) *Introspection in Second Language Research*, pp. 24–54. Multilingual Matters, Clevedon, Avon (1987)
21. Ericsson, K., Simon, H.: *Protocol Analysis: Verbal Reports as Data*, 2nd edn. MIT Press, Boston (1993)
22. Fall, J., Fall, A.: SELES: A Spatially Explicit Landscape Event Simulator. In: Proceedings of GIS and Environmental Modeling, pp. 104–112. National Center for Geographic Information and Analysis (1996)
23. Forlines, C., Shen, C., Wigdor, D., Balakrishnan, R.: Exploring the effects of group size and display configuration on visual search. In: *Computer Supported Cooperative Work 2006 Conference Proceedings*, pp. 11–20 (2006)
24. Garfinkel, H.: *Studies in Ethnomethodology*. Polity Press, Cambridge (1967)

25. Gonzalez, V., Kobsa, A.: A Workplace Study of the Adoption of Information Visualization systems. In: Proceedings of the International Conference on Knowledge Management, pp. 92–102 (2003)
26. Gorard, S.: Combining Methods in Educational Research. McGraw-Hill, New York (2004)
27. Greenberg, S., Buxton, B.: Usability Evaluation Considered Harmful (Some of the Time). In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (2008)
28. Greene, J., Caracelli, V., Graham, W.: Toward a Conceptual Framework for Mixed-Method Evaluation Design. *Educational Evaluation and Policy Analysis* 11(3), 255–274 (1989)
29. Gutwin, C., Greenberg, S.: The Mechanics of Collaboration: Developing Low Cost Usability Evaluation Methods for Shared Workspaces. In: Proceedings WETICE, pp. 98–103. IEEE Computer Society Press, Los Alamitos (2000)
30. Healey, C.G.: On the Use of Perceptual Cues and Data Mining for Effective Visualization of Scientific Datasets. In: Proceedings of Graphics Interface, pp. 177–184 (1998)
31. Heer, J., Viegas, F., Wattenberg, M.: Voyagers and Voyeurs: Supporting Asynchronous Collaborative Information Visualization. In: Proceedings of the Conference on Human Factors in Computing Systems (CHI'07), pp. 1029–1038. ACM Press, New York (2007)
32. Holtzblatt, K., Beyer, H.: Contextual Design: Defining Customer-Centered Systems. Morgan Kaufmann, San Francisco (1998)
33. Huck, S.W.: Reading Statistics and Research, 4th edn. Pearson Education Inc., Boston (2004)
34. Interrante, V.: Illustrating Surface Shape in Volume Data via Principal Direction-Driven 3D Line Integral Convolution. *Computer Graphics, Annual Conference Series*, pp. 109–116 (1997)
35. Isenberg, P., Carpendale, S.: Interactive Tree Comparison for Co-located Collaborative Information Visualization. *IEEE Transactions on Visualization and Computer Graphics* 12(5) (2007)
36. Isenberg, P., Tang, A., Carpendale, S.: An Exploratory Study of Visual Information Analysis. In: Proceedings of the Conference on Human Factors in Computing Systems (CHI'08), ACM Press, New York (to appear, 2008)
37. Kay, J., Reiger, H., Boyle, M., Francis, G.: An Ecosystem Approach for Sustainability: Addressing the Challenge of Complexity. *Futures* 31(7), 721–742 (1999)
38. Kim, S., Hagh-Shenas, H., Interrante, V.: Conveying Shape with Texture: Experimental Investigations of Texture's Effects on Shape Categorization Judgments. *IEEE Transactions on Visualization and Computer Graphics* 10(4), 471–483 (2004)
39. Kleffner, D.A., Ramachandran, V.S.: On the Perception of Shape from Shading. *Perception and Psychophysics* 52(1), 18–36 (1992)
40. Kobsa, A.: User Experiments with Tree Visualization Systems. In: Proceedings of the IEEE Symposium on Information Visualization, pp. 9–26 (2004)
41. Kruger, R., Carpendale, S., Scott, S.D., Greenberg, S.: Roles of Orientation in Tabletop Collaboration: Comprehension, Coordination and Communication. *Journal of Computer Supported Collaborative Work* 13(5–6), 501–537 (2004)
42. Lewin, C. (ed.): *Research Methods in the Social Sciences*. Sage Publications, London (2004)
43. Lewis, C., Rieman, J.: *Task-Centered User Interface Design: A Practical Introduction* (1993)
44. Likert, R.: A Technique for the Measurement of Attitudes. *Archives of Psychology* 140, 1–55 (1932)
45. Lofland, J., Lofland, L.: *Analyzing Social Settings: A Guide to Qualitative Observation and Analysis*. Wadsworth Publishing Company, CA, USA (1995)
46. Mankoff, J., Dey, A., Hsieh, G., Kientz, J., Lederer, S., Ames, A.: Heuristic Evaluation of Ambient Displays. In: Proceedings of CHI '03, pp. 169–176. ACM Press, New York (2003)
47. Mark, G., Kobsa, A., Gonzalez, V.: Do Four Eyes See Better Than Two? Collaborative Versus Individual Discovery in Data Visualization Systems. In: Proceedings of the IEEE Conference on Information Visualization (IV'02), July 2002, pp. 249–255. IEEE Press, Los Alamitos (2002)

48. Mazza, R., Berre, A.: Focus Group Methodology for Evaluating Information Visualization Techniques and Tools. In: Proceedings of the International Conference on Information Visualization IV (2007)
49. McCarthy, D.: Normal Science and Post-Normal Inquiry: A Context for Methodology (2004)
50. McGrath, J.: Methodology Matters: Doing Research in the Social and Behavioural Sciences. In: Readings in Human-Computer Interaction: Toward the Year 2000, Morgan Kaufmann, San Francisco (1995)
51. Moggridge, B.: Design Interactions. MIT Press, Cambridge (2006)
52. Morris, M.R., Ryall, K., Shen, C., Forlines, C., Vernier, F.: Beyond “Social Protocols”: Multi-User Coordination Policies for Co-located Groupware. In: Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW, Chicago, IL, USA), CHI Letters, November 6-10, 2004, pp. 262–265. ACM Press, New York (2004)
53. Morse, E., Lewis, M., Olsen, K.: Evaluating Visualizations: Using a Taxonomic Guide. *Int. J. Human-Computer Studies* 53, 637–662 (2000)
54. Nielsen, J., Mack, R.: Usability Inspection Methods. John Wiley & Sons, Chichester (1994)
55. North, C.: Toward Measuring Visualization Insight. *IEEE Computer Graphics and Applications* 26(3), 6–9 (2006)
56. Patton, M.Q.: Qualitative Research and Evaluation Methods, 3rd edn. Sage Publications, London (2001)
57. Plaisant, C.: The Challenge of Information Visualization Evaluation. In: Proceedings of the Working Conference on Advanced Visual Interfaces, pp. 109–116 (2004)
58. Purchase, H.C., Hoggan, E., Görg, C.: How Important Is the “Mental Map”? – An Empirical Investigation of a Dynamic Graph Layout Algorithm. In: Kaufmann, M., Wagner, D. (eds.) GD 2006. LNCS, vol. 4372, pp. 184–195. Springer, Heidelberg (2007)
59. Purchase, H.C.: Effective Information Visualisation: A Study of Graph Drawing Aesthetics and Algorithms. *Interacting with Computers* 13(2), 477–506 (2000)
60. Purchase, H.C.: Performance of Layout Algorithms: Comprehension, Not Computation. *Journal of Visual Languages and Computing* 9, 647–657 (1998)
61. Brandenburg, F.J. (ed.): GD 1995. LNCS, vol. 1027. Springer, Heidelberg (1996)
62. Reilly, D., Inkpen, K.: White Rooms and Morphing Don’t Mix: Setting and the Evaluation of Visualization Techniques. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 111–120 (2007)
63. Saraiya, P., North, C., Duca, K.: An Insight-Based Methodology for Evaluating Bioinformatics Visualizations. *IEEE Transactions on Visualization and Computer Graphics* 11(4), 443–456 (2005)
64. Scott, S.D., Carpendale, S., Inkpen, K.: Territoriality in Collaborative Tabletop Workspaces. In: Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW, Chicago, IL, USA), CHI Letters, November 6-10, 2004, pp. 294–303. ACM Press, New York (2004)
65. Seidman, I.: Interviewing as Qualitative Research: A Guide for Researchers in Education and the Social Sciences. Teachers’ College Press, New York (1998)
66. Shneiderman, B.: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In: Proceedings of the IEEE Symposium on Visual Languages, pp. 336–343. IEEE Computer Society Press, Los Alamitos (1996)
67. Shneiderman, B., Plaisant, C.: Strategies for Evaluating Information Visualization Tools: Multi-Dimensional In-Depth Long-Term Case Studies. In: Proceedings of the Workshop on Beyond Time and Errors: Novel Evaluation Methods for Information Visualization, BELIV (2006)
68. Spence, R.: Information Visualization, 2nd edn. Addison-Wesley, Reading (2007)
69. Spool, J., Schroeder, W.: Testing Web Sites: Five Users is Nowhere Near Enough. In: CHI ’01 Extended Abstracts, pp. 285–286. ACM Press, New York (2001)

70. Strauss, A.L., Corbin, J.: *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Sage Publications, London (1998)
71. Tang, A., Tory, M., Po, B., Neumann, P., Carpendale, S.: Collaborative Coupling over Tabletop Displays. In: *Proceedings of the Conference on Human Factors in Computing Systems (CHI'06)*, pp. 1181–1290. ACM Press, New York (2006)
72. Tang, A., Carpendale, S.: An observational study on information flow during nurses' shift change. In: *Proc. of the ACM Conf. on Human Factors in Computing Systems (CHI)*, pp. 219–228. ACM Press, New York (2007)
73. Tang, J.C.: Findings from observational studies of collaborative work. *International Journal of Man-Machine Studies* 34(2), 143–160 (1991)
74. Tory, M., Möller, T.: Evaluating Visualizations: Do Expert Reviews Work. *IEEE Computer Graphics and Applications* 25(5), 8–11 (2005)
75. Tufte, E.: *The Visual Display of Quantitative Information*. Graphics Press, Cheshire (1986)
76. Tufte, E.: *Envisioning Information*. Graphics Press, Cheshire (1990)
77. Tufte, E.: *Visual Explanations. Images and Quantities, Evidence and Narrative*. Graphics Press, Cheshire (1997)
78. Viegas, F.B., Wattenberg, M., van Ham, F., Kriss, J., McKeon, M.: Many Eyes: A Site for Visualization at Internet Scale. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization / Information Visualization 2007)* 12(5), 1121–1128 (2007)
79. Ware, C.: *Information Visualization: Perception for Design*, 2nd edn. Morgan Kaufmann, San Francisco (2004)
80. Wigdor, D., Shen, C., Forlines, C., Balakrishnan, R.: Perception of Elementary Graphical Elements in Tabletop and Multi-surface Environments. In: *Proceedings of the Conference on Human Factors in Computing Systems (CHI'07)*, pp. 473–482. ACM Press, New York (2007)
81. Willett, W., Heer, J., Agrawala, M.: Scented Widgets: Improving Navigation Cues with Embedded Visualizations. In: *INFOVIS 2007. IEEE Symposium on Information Visualization (2007)*
82. Yost, B., North, C.: The Perceptual Scalability of Visualization. *IEEE Transactions on Visualization and Computer Graphics* 12(5), 837–844 (2006)
83. Zuk, T.: *Uncertainty Visualizations*. PhD thesis. Department of Compute Science, University of Calgary (2007)
84. Zuk, T., Carpendale, S.: Theoretical Analysis of Uncertainty Visualizations. In: *Proceedings of SPIE Conference Electronic Imaging, Vol. 6060: Visualization and Data Analysis (2006)*
85. Zuk, T., Schlesier, L., Neumann, P., Hancock, M.S., Carpendale, S.: Heuristics for Information Visualization Evaluation. In: *Proceedings of the Workshop BEyond Time and Errors: Novel Evaluation Methods for Information Visualization (BELIV 2006)*, held in conjunction with the Working Conference on Advanced Visual Interfaces (AVI 2006), ACM Press, New York (2006)

# Many Eyes: A Site for Visualization at Internet Scale

Fernanda B. Viégas, Martin Wattenberg, Frank van Ham, Jesse Kriss, Matt McKeon

**Abstract**— We describe the design and deployment of Many Eyes, a public web site where users may upload data, create interactive visualizations, and carry on discussions. The goal of the site is to support collaboration around visualizations at a large scale by fostering a social style of data analysis in which visualizations not only serve as a discovery tool for individuals but also as a medium to spur discussion among users. To support this goal, the site includes novel mechanisms for end-user creation of visualizations and asynchronous collaboration around those visualizations. In addition to describing these technologies, we provide a preliminary report on the activity of our users.

**Index Terms**—Visualization, World Wide Web, Social Software, Social Data Analysis, Communication-Minded Visualization.

---

## 1 INTRODUCTION

When visualization researchers talk about scaling, we usually mean handling large data sets. We seek ways to draw huge graphs, explore high dimensional spaces, and display databases with billions of rows. In this paper, however, we consider an alternate perspective: Instead of scaling the size of the data, what happens when we scale the size of the audience?

This perspective is suggested by the rise of the Web as a visualization platform. Recent years have witnessed internet-based visualizations ranging from political art projects (e.g. Theyrule [25]) to New York Times stories (Faces of the Dead [16]). These displays are seen by thousands and it is natural to ask what new opportunities arise when visualizations move to an environment where vast crowds of people can create, view and discuss them. Not only are interactive visualizations a key medium for communication in a data-rich world, but preliminary reports hint that visualizations potentially have a catalytic effect on storytelling [26] and collective data analysis [28].

Unfortunately there are two main roadblocks to overcome before visualization has a chance to fulfil this potential. First, the creation and publishing of interactive visualizations remains accessible only to specialists. While frameworks such as Flash and Processing ease development, they do not help non-programmers. End user tools for sophisticated visualizations such as Tableau [23] and Spotfire [21] still require expertise, installation and training. Furthermore, none supports easy publishing to the public web. A non-developer blogger who wishes to write about an interesting data set, for instance, is currently limited to static charts.

Even if that hypothetical blogger manages to publish an interactive visualization, a second challenge remains: how can readers discuss it? Web-based visualization has obvious social and collaborative potential, but without special technology it can be hard to have a discussion around a visualization. If one person sees something interesting, for instance, how can they point it out to others? Systems that do explicitly support collaboration, e.g., DEVis [14], Spotfire [21], and the Command Post of the Future [20], have been aimed at scientists and other experts operating in a closed intranet or military environment. The open Web is a very different environment: the designers of the Command Post of the Future likely did not want to encourage public discussion about the discoveries its users made.

This paper describes a public web site, Many Eyes, that addresses the challenges of end user construction and the unique environment of open web-based collaboration.

- 
- All authors are with IBM Research. E-Mail: viegasf@us.ibm.com, mwatten@us.ibm.com, fvanham@us.ibm.com, jesse.kriss@us.ibm.com, mmmckeon@us.ibm.com respectively.

Manuscript received 31 March 2007; accepted 1 August 2007; posted online 27 October 2007.

For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

The site launched on January 23, 2007 and allows users to upload data, create visualizations of that data, and leave comments on both visualizations and data sets. In addition to constituting a research platform, the site is an effort to “democratize” visualization technology. By this we mean both providing the technology to the broadest possible audience and fostering a democratic deliberative style of data analysis.

One lesson learned from building this site is that the most difficult issues in scaling the audience are not necessarily related to engineering. Instead, many of the key questions are those of design and user experience. To give one example of the difference between Many Eyes and a system such as Tableau or Spotfire, consider that many of our users arrive directly at our visualizations for the first time via links from external web sites. No one is an accidental user of a commercial system such as Tableau; by contrast, our audience often views the site with little context and absolutely no training. If they do not understand what they see, they can (and will) simply click away from the site, never to return.

This fact of life on the web implies a goal of instant usability that has ramifications for all aspects of the site. The data model we support, features offered to users, and flavors of visualization techniques supported on Many Eyes all reflect the need for broad, immediate accessibility. These implications are reflected in design choices that need to strike the delicate balance between powerful data-analysis capabilities and accessibility to the non-expert visualization user.

Our aim in this paper is to describe the design of the site and the rationale behind the decisions we made. We will focus on decisions that affected our two overarching goals: enabling end-user creation of visualizations and fostering large-scale collaborative usage. The design decisions in Many Eyes fall into three different areas: information visualization, end-user data collection and manipulation, and the social aspects of collaborative analysis. The organization of the paper reflects this division. Section 2 is an overview of the Many Eyes web site and its basic functions. Section 3 presents the data considerations in the site. Sections 4 and 5 cover the different kinds of visualizations and the social/collaborative features available. Finally, Section 6 presents preliminary results on usage. Since the work introduced here spans a variety of disciplines, related work is discussed in multiple sections, each one addressing a specific area of research.

### 1.1 Related Work—*asynchronous collaborative visualization*

There is a large body of work on collaboration in visualization systems [2][14][20][27]. Long-standing research has explored techniques for collocated collaboration (e.g., large displays and shared workspaces) and synchronous distance work (e.g., real-time shared visual exploration environments). However, less research



attention has been devoted to asynchronous collaboration around visualizations or to the questions that arise in mass-audience visualizations.

Sense.us [12] is probably the research project that comes closest in spirit to Many Eyes. It is a web-based prototype that supports commenting and annotations on visualizations of US Census data. The system provides mechanisms for facilitating view sharing, threaded discussion across views, and the construction of tours and presentations. Unlike Many Eyes, Sense.us was pre-populated with six visualizations; users could not contribute new data sets or create new visualizations. The system was deployed on a corporate intranet, as opposed to the public web and was designed to explore issues of discussion and annotation rather than content collection and creation around visualization. With a small number of visualizations produced by the site administrator and its presence in corporate environment, Sense.us felt very different from Many Eyes, which contains thousands of visualizations and data sets contributed by users, spanning a wide array of serious and non-serious themes.

A number of commercial systems have begun to explore the idea of asynchronous communication around data. Several web sites (e.g., Dataplace [9], Data360 [8], Swivel [22], DabbleDB [7] and Chartall [5]) allow users to upload and graph data. Swivel, Data360, and Chartall allow users to make comments; Swivel especially seems to aim at scaling to a large audience, styling itself as “YouTube for data”. All of these rely on static standard business graphics, however, and so do not tackle issues of state bookmarking and end-user construction that arise in the context of sophisticated interactive visualizations.

DEVise [14] is an early exploration into the benefits of sharable visualizations. It offers both customizable visual mappings and sharable views, as well as basic annotation functionality. Although designed to run in a browser it was not designed to be publicly accessible and most of its visualizations were relatively static.

The commercial Spotfire [21] system does feature sophisticated visualizations and includes a product called DecisionSite Posters, a web-based system that enables users to create bookmarks and comments attached to specific views of a visualization. While impressive, the “posters” are essentially an adjunct to the main desktop Spotfire application. Because this product is aimed at intranet usage by expert analysts, it does not explore the design issues involved in scaling to mass audiences on the public web.

## 2 A BRIEF OVERVIEW OF THE MANY EYES SITE

Many Eyes is roughly modeled on well-known participatory sites such as Flickr [10] and YouTube [30]. The central activities on the site are to upload data, construct visualizations, and leave comments on either data sets or visualizations. Each of these activities is described in detail in subsequent sections.

To navigate the continuously growing collections of visualizations, data sets, and comments, the site contains “browsing” pages that display recent contributions. The data set browsing page, for instance, shows a table with the latest data sets to have been uploaded to the site. The table also displays metadata about each data set: keywords (“tags”), source, the contributor’s username, size in bytes and number of rows, date of contribution and links to existing visualizations with that data set (Fig 1).

The browse pages are not the only navigational aids. Every day the home page prominently features four visualizations that are chosen by the designers of the site and typically reflect newsworthy events or good models of visualization usage. For users who are interested in a specific topic, we provide a simple text based search function for both data sets and visualizations.

All visualizations and data sets on Many Eyes have an attached discussion forum where users can share textual comments and links to other webpages (Fig. 4). Since all content on Many Eyes resides at a fixed URL, users can also easily link to other visualizations on the site from both inside and outside Many Eyes..

After the site launched in January 2007, it was featured in several prominent blogs as well as some mainstream media outlets. This publicity has provided a steady stream of visitors to the site.

## 3 DATA

Data is at the root of all activity on Many Eyes. Although we seeded the site with a few data sets and visualizations, most of the content is contributed by ordinary users, who upload data sets, discuss them, create visualizations from the data, and then discuss those visualizations.

The fact that users can upload their own data offers a number of potential benefits. The benefits to individuals are clear, since they can visualize their own data. One might also hope for a collective benefit: a user might upload one set of data, and then another user could augment it by uploading additional, related information. Finally, from the point of view of research, this is a unique opportunity to understand user demand: what types of data do people really want to visualize?

Gaining these benefits, however, means balancing complex and conflicting constraints on the design. The fact that data is uploaded to the site by end users means that the data model needs to be easily understandable, with a format that is appropriate for non-programmers. At the same time, the format needs to be flexible enough to express the data structures used by visualizations such as treemaps and graph layout algorithms.

In addition to constraints on the data model, we face constraints on changes to data. Data sets and their visualizations are subjects of discussion so that comments and annotations may go stale if the underlying data is edited. A second difficulty is that the kind of data “reshaping” that is often necessary in preparing a data set for visualization can be difficult to explain to lay users without the visual context.

### 3.1 The data model

The core data model used by Many Eyes is a table: that is, a set of named columns, each of the same length. Each column has a type, which currently can be either text or numeric. The site also supports data that comes in the form of unstructured text, i.e. a sequence of characters, equivalent to a Java String or a CLOB (“character large object”) in a relational database.

Each data set, whether a table or unstructured text, is associated with a collection of metadata. Some metadata, such as a (required) title, the source of the data and a paragraph-length description is provided by the user. Other metadata is automatically set by the system, such as creation date and author. Datasets are currently stored on our servers in plain text format, a set-up that has proven to be efficient and simple.

### 3.2 Uploading data

Users upload data via an HTML form. The form contains a text area where the user can paste in a data set. The data can either be freeform text, in which case it is interpreted as unstructured data, or a tab-delimited grid, in which case it is interpreted as a table. We chose to use tabs as delimiters because it allows users to simply copy and paste from Microsoft Excel or Open Office.

In the case of tabular data, the system makes a guess about whether each column is numeric or textual, using heuristics that account for currency symbols and the difference between US or European punctuation. In the event that the system makes a mistake, the user may override the automatic type choice.

As in the visualization-creation step, described later, user education is an important consideration: after all, many users will not have experience in preparing data so that it is understandable by others. The user is encouraged to provide as much metadata as possible and we include a prominent reminder to label units. Our “help” page describes not just the technical details of the data format, but a section on the “elements of data style” that guides users in creating data that will be comprehensible by others.

Data	Keywords	Source	Contributor	Size	Added	Existing Visualizations	Visualize	Edit	Delete
PhageOType by HostProtein	bacteriophage genomics	Fraser et al	JFraser	1% 53 rows	Friday, 4:46 PM				
the litter bin			ClubAM	109% 2946 rows	Friday, 3:51 PM				
secondlife stats	economics instablab	Linden Lab blog - Key Me...	Roo	2% 102 rows	Friday, 3:31 PM				
Number of Hits and Visits by Search Query (March...		Gallery of Art & Design...	blustena9	1% 51 rows	Friday, 2:47 PM				
Tag Cloud Instructions for creating a Tag Cloud V...			Nebulous	5k	Friday, 2:32 PM				
SLIRC Race 11 stroke data	Concept2 Indoor Rowing stroke data	Concept2 venue racing so...	Dougie	142k 7730 rows	Friday, 2:19 PM				
Utility Supplier			blueBoy	445 bytes 11 rows	Friday, 1:34 PM				
Stability vs Noise	Personal		Frank van Ham	5% 52 rows	Friday, 1:21 PM				

Fig. 1. The data browse page on Many Eyes, giving an overview of all datasets currently uploaded to the site.

A key decision was whether to require registration for users to upload data to the site. Registration is a significant barrier to entry, so on its face it would seem to work against our desire to reach a broad audience. For two reasons, however, we decided to require registration. First, we wanted to provide a slight barrier to entry to prevent frivolous or malicious data uploads. Second, during the registration process we could require explicit agreement to certain legal terms.

Finally, users can navigate the collection of data sets in a variety of ways: a standard search box; use of keywords or “tags”; and a list that can be ordered by upload data, contributor, and other meta data.

### 3.3 Working with data

Each data set that is uploaded to the site is given its own page. This page contains metadata, a snippet of the data set, and a discussion forum on which users can talk about the information. We also provide a link to the original version of the data in plain text form. At the bottom of the data page is a button that starts the process of matching the data with a visualization (see section 4.2) as well as a set of small icons that indicate which methods have been used to visualize the data already.

As mentioned above, Many Eyes currently does not allow any direct editing of the data sets: each is immutable. The reason is that editing data could invalidate any visualizations created from the data set as well as comments made on those visualizations. This was a difficult decision to make, however, since users clearly would benefit from being able to edit data: not only have we received numerous requests for this feature, but we have seen users upload many separate versions of a file as they correct errors. We do currently allow users to delete their own data sets if no visualizations have been made from them.

### 3.4 Related work—data representation

Several online data storage services aimed at end users exist already. Two examples are Intuit’s Quickbase [18] and DabbleDB [7]. Quickbase uses simple tables, much like Many Eyes, but with a somewhat richer set of types. DabbleDB is more akin to a database and also offers basic visualization features. As mentioned in the introduction, several other social data analysis sites have appeared contemporaneously with Many Eyes, including Data360, Swivel, and Chartall. All using similar tabular formats, although none allow the kind of data reshaping described in Section 4.

Why have these sites converged on this solution? One answer is that all want to exploit the vast amount of data stored in Microsoft Excel spreadsheets. Furthermore, tables are simple and well-understood by end users.

Two alternatives that we considered were to expose a standard relational database model or an OLAP data cube. While this would make it easy to extend Many Eyes to interface with standard data stores (as with Spotfire or Tableau) it seemed unlikely to appeal to end users. Unfortunately, even the ability to use simplified systems such as Microsoft Access is quite rare. Similarly, we judged data cubes to be beyond the knowledge of a typical user: The closest thing to a data cube that most people have encountered is a Microsoft Excel pivot table, and informal discussions with potential users uncovered a lack of understanding of (even outright hostility toward) this tool.

A similar problem exists with semi-structured data models based on XML. Although flexible and friendly to developers, an XML-based model seemed likely to be a barrier for nonprogrammers. For example we asked a professor who regularly teaches introductory statistics if she thought importing XML-formatted data would be important to Many Eyes; her answer was, “What’s XML?”

## 4 VISUALIZATIONS

Many Eyes relies on a pure web-based model in order to reach the largest possible audience. While visualizations have existed on the web for more than a decade, these have been constructed offline and then separately published to the web. On Many Eyes, however, visualizations can be constructed and published by users without ever leaving the browser. Although a browser-based environment imposes some constraints, it is critical to the goal of scaling our audience.

The visualization technology in Many Eyes consists of a set of individual components that represent different display techniques. To create a visualization, a user combines a component with a data set, in a manner described below. Once created, the visualization is given its own home page, which also contains metadata, a link to the original data set, and an area for discussion. Each page carries a prominent link to a compact description that explains the visualization technique being used, what its data requirements are and in what cases it can best be applied. Explanation pages may also contain “expert notes,” which usually address some of the subtleties of using a specific technique over another—for example, when one should use a bar chart over a line graph.

### 4.1 Visualization components

Many Eyes provides more than a dozen different types of basic visualization components (see Fig. 2 for a small taste). The menu includes a mix of standard business graphics (e.g., bar charts), well-established academic techniques (e.g., treemaps), and a few experimental components (bubblecharts and tag clouds). Each of these visualization components was implemented as a separate java applet to keep download sizes to a minimum. Since we lack the space to discuss every single applet separately, we will suffice with a short overview of some of our more experimental visualizations.

#### 4.1.1 Bubble charts

One of the most common types of real world data is simply a list of labeled numbers. Movie titles and box office grosses, for instance, or basketball players and salaries. Such lists are often easily displayed via standard bar charts or histograms, but these methods can run into trouble for highly skewed power-law distributions. To address this difficulty, we included a “bubble chart” which represents items by labelled circles, whose areas are proportional to the displayed quantity. These charts effectively perform a visual square-root transform.

#### 4.1.2 Stack graph for categories

Many time series have a hierarchical structure: the history of the United States federal budget, for example, can be recursively subdivided into various departmental levels: defense overall, atomic weapons, and so forth. To display this type of data, we included a “stack graph for categories” visualization, based on the

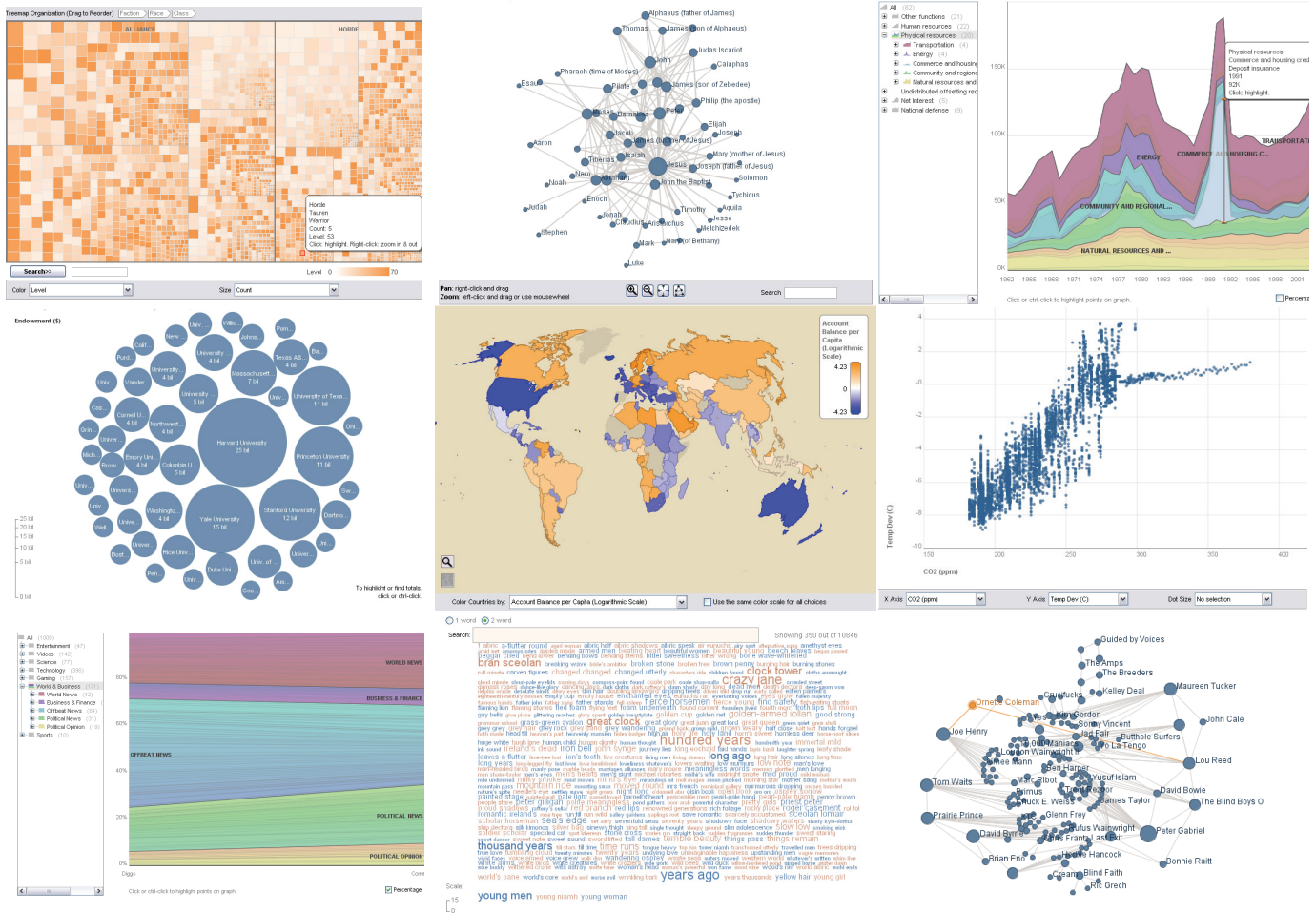


Fig. 2. A tiny cross-section of the visualizations users created by matching their own data with our visualization components. From left to right and top to bottom: A treemap on character data in a World of Warcraft realm, a network diagram of social relationships in the bible, a stack graph showing the categorized spending of the US government over time, a bubble chart showing the size of the endowments of different US universities, a world map showing the average account balance per capita, a scatterplot comparing CO2 levels against temperature, a cleverly used stacked graph that shows relative differences in the type of postings on two online link aggregators, a tag cloud showing two word occurrences in the work of Yeats and a network diagram of musical ties.

BookVoyager design described in [29]. To the left of the graph is a tree control that follows the hierarchical structure of the time series. Clicking on elements in the tree control filters the stack graph to show only time series at that level of hierarchy. The icon representing each hierarchy level is a tiny “sparkline” graph that provides a preview of what the user will see upon clicking. This helps new users understand how the control works, as well as providing an overview of the individual series.

### 4.1.3 Graph drawing

Although network data can be found in many different application areas, there are surprisingly few online tools that can provide a graph visualization of a user uploaded file. The Graph Drawing Server [3] was one of these services, but seems to be inactive now. The Large Graph Layout [1] server is another but only returns a list of coordinates by e-mail, which can hardly be considered interactive. We have implemented a force directed graph drawing algorithm in a zoomable user interface. The layout algorithms (although implemented in java) are able to provide layouts for graphs up to a thousand nodes in a couple of seconds, and only have to run when the visualization is created. Although many different ‘standardized’ dataformats for graphs exist, we settled on a simple edge table. Although not the most efficient storage format it proved both highly understandable to the end users, and fitted nicely with our table oriented design.

### 4.1.4 Tag clouds

Since one of the types of data that users frequently uploaded were word counts, we decided to include a visualization that would take unstructured text as input. Tag clouds can quickly give the user an overview of the most salient terms in a large corpus of text. We implemented a number of improvements over the standard tag cloud applets. One of these is the ability to measure the frequency of two word tags in the text. The other includes the ability to dynamically filter the tag cloud by entering query strings in a text box. We found tag clouds attracting a whole new set of users, whose interest is primarily in textual data instead of numbers.

## 4.2 Visual mapping

A visualization is created by matching a dataset with a visualization component. Of course not all visualizations display the same type of input data. A treemap, for example, requires a number of textual columns to define its hierarchy and two numerical columns that map to size and color. On the other hand, a basic scatterplot requires two numerical columns (one for each axis), an optional numerical column that specifies the size of each dot and a textual column for the labels. A single datafile might be used to drive both visualizations, offering different perspectives on the data (Figure 3). To set up this mapping, the visualization components need to be able to express its data needs in a precise manner. In Many Eyes, a component’s data needs are expressed in a schema specified by the

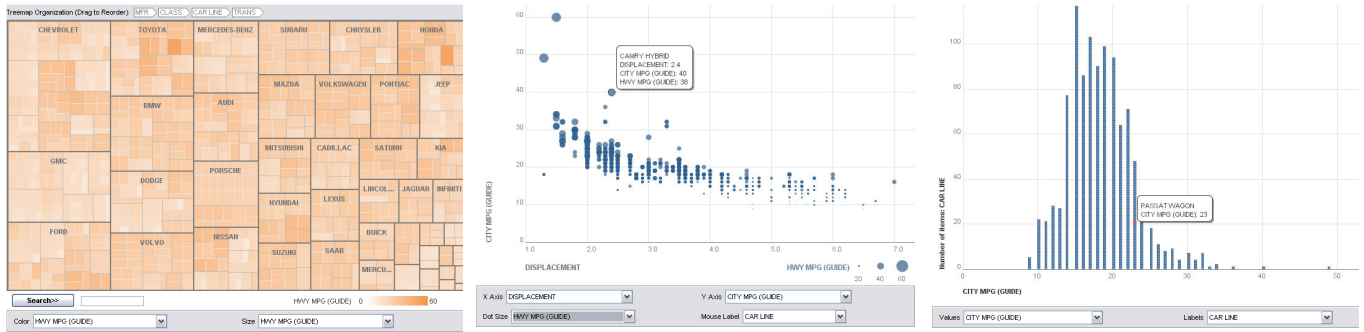


Fig. 3. Three user generated visualizations offering different perspectives on the same dataset on car fuel economy. The grey areas on the top and bottom are automatically generated by the application and allow the user to browse through different dimensions in the data.

visualization programmer. A schema is a set of named, typed slots. Currently we support 5 different slot types:

- **Unstructured slots (U)**: flat unstructured text
- **Numeric slots (N)**: a single column of numeric data.
- **Textual slots (T)**: a single column of textual data.
- **Multiple numeric slots (N<sup>+</sup>)**: one or more columns of numeric data.
- **Multiple textual slots (T<sup>+</sup>)**: one or more columns of textual data.

Using these 5 basic slot types we can express the data needs for each visualization component. Hierarchies on the data can be expressed as an ordered set of textual columns, where each row in the set describes the path from the top of the hierarchy to the leaf item. We can then express the schemas of a treemap and a scatterplot in the following manner, respectively : {hierarchy : T<sup>+</sup>, size : N, color : N } and {Xaxis : N, Yaxis : N, label : T, [Dotsize : N]}.

Slots can be made optional, so that no mapping is necessary for a visualization to be complete. For example, our scatterplot includes an optional slot for dot size, while the slots for x- and y-axes are required. The challenge in matching up a visualization technique with a dataset is to map the correct columns in the data table to a slot. Since both slots and columns are typed we can make some inference on potential matches. For Many Eyes, we decided on a mix of automated and user generated mapping, depending on the circumstances. For slots that can accept multiple columns (i.e. slot types N<sup>+</sup> and T<sup>+</sup>) we feed the visualization all columns in the table that have a matching type. For slots that only take a single column, we decided to partly leave this mapping to the user and allow him or her to choose from a number of type compatible columns.

### 4.3 End User Data Manipulation

One option is to allow the actual viewers of the final visualization to change the mapping, making it possible to change the data shown in the visualization on the fly. This offers a fast way to browse through different dimensions on a dataset. For example, matching a single numerical table with a scatterplot allows the end user to select two

columns to display on the x and y axes from the collection of all possible columns. The user interface for these selections is generated automatically from the mapping. For each slot where multiple columns might be a valid match we generate a drop down box below the visualization (see Fig. 3). However, depending on the visualization technique it might be very well possible that not every selection in the drop down box will produce meaningful visualization results. Take the world map as an example: it requires a text column as input for the location slot (typically this contains the country name), but if we have multiple text columns in the data, generally only one will produce a meaningful result. In this case the selection has to be done by the person creating the initial visualization, after which it cannot be changed by viewers. The choice between end user selection and creator selection is specified by the visualization programmer for each slot.

#### 4.3.1 Contextual Data Transformation

The ability to change mapping of columns to visual attributes is not the only data manipulation option we included. As described in Section 3, there are many types of data transformation or reshaping that users may need to perform. In many data acquisition pipelines these transformations occur in a separate stage. Due to our end-user audience we have opted for *contextual data transformation*, that is, we let users perform all such transformation in the context of creating a visualization, so that they may easily see and understand the results of their actions.

For example, in several visualization components it may make sense to transpose the rows and columns of the input data table. Rather than asking the user to perform this operation before starting to visualize their data, a “Flip rows/columns” button is made available whenever the column types permit the operation.

In some cases it might also be useful for the end user to be able to reorder the columns that were fed into a multiple column slot (i.e. N<sup>+</sup> or T<sup>+</sup>). For example, multiple textual columns can define a hierarchy on items, but the user might want to reorder them to get different orders of aggregation. We designed a widget that allows users to reorder column names by drag and drop, again, changing the visualization on the fly (see leftmost sample in Figure 3).

A more subtle type of manipulation relates to rolling up data sets. In early trials it became clear that users expected a certain kind of automatic aggregation. For example, imagine a bar chart of basketball salaries, where the columns in the underlying data set are player name, position, and salary. When the label slot was set to player, the bar chart yields—as expected—a chart of individual player salaries. However, when the label slot was set to position test users indicated that they expected the bars for each position to be aggregated, preferably by averaging. Interestingly, users did not always expect averaging. In some bar charts they wanted summation rather than averaging. In pie charts—which are designed to show relative totals—it seemed that aggregation should always occur by summation. And in a scatterplot, users did not expect any aggregation! To handle these expectations, we created a set of

Table 1: Available Visualization Types in Many Eyes

Technique	Data schema
Bubblechart	
Histogram	
Pie Chart	{Labels / item names : T, Values : N}
Maps	
Tag Cloud	
Bar chart	
Line graph	{Axis labels : T, Values : N+}
Stack graph	
Network diagram	{From : T, To : T}
Scatterplot	{Xaxis : N, Yaxis : N, Label: T, [Dotsize : N]}
Stack graph/categories	{Hierarchy : T+, Values : N+}
Treemaps	{Hierarchy : T+, Size : N, Color : N}
Tag Cloud	{U}

aggregation widgets with customizable default actions for the different visualization components.

One last example of data normalization relates to the maps. While U.S. state names are fairly standardized, names of countries can appear in many different ways: e.g., “Democratic People’s Republic of Korea” “Korea, People’s Republic,” or simply “North Korea.” When one of our map components does not recognize a name, it uses a simple distance measure to suggest a likely match, while allowing users to override this match as necessary.

#### **4.4 Related work—end user construction of visualizations**

Visualization component models that include a notion of mapping table columns to visual attributes are not new—in fact they may be better characterized as a known best practice in the field. At the simplest level, Microsoft Excel generates graphs by letting users select columns that feed into business charts. The well-known work of [15], parameterized the different visual encodings in visualizations, using them to automatically choose a meaningful visual representation for a given dataset. Some frameworks [19][14] completely parameterize the visualizations and map directly between a data tuple and the shape and position of its representation on screen. More flexible end user desktop visualization applications include Spotfire [21] and Tableau [23], which both offer advanced data mapping paradigms. Many Eyes may be most similar to the systems of [17] and [24] that provide a number of commonly used visualizations with predefined slots and map data tuples to slots.

Where Many Eyes differs from existing end-user visualization systems is the pure web-based interaction and publishing model, which makes visualization construction tools immediately available to millions of people with internet access. While Spotfire allows the publishing of “posters,” these must be created with the desktop application and represent a subset of the desktop functionality. In addition, the contextual data transformation approach distinguishes Many Eyes from systems that include a separate data reshaping stage. Finally, the palette of visualization components and their design reflects the need to provide instant utility to users on a broad range of data sets.

## **5 SOCIAL FEATURES**

So far we have concentrated on the constraints of an open web platform, and described the tradeoffs necessary to meet them. In this section we discuss some of collaborative features that exploit the opportunities of an open web deployment. In particular, we describe how we allow users to engage their collective intelligence, by pointing to items of interest, sharing insights, asking questions, and monitoring activity on items of interest.

Previous systems have explored such capabilities, but as we discuss below the web provides a unique social environment. One important distinction is that communication around visualizations can potentially occur both on and off site. Thus, users should not be restricted to discussing Many Eyes visualization only on the site. For this reason, it became important to provide points of entrance to discussions that were external to the site itself—for instance on blogs or in forums.

### **5.1 On-site communication**

The main communication feature in Many Eyes is the textual comment. Comments exist in the context of specific visualizations and data sets. As users interact with a visualization, they can enter comments very much in the same way comments are entered into a blog. The same is true of data sets; each data set has a page where comments may be entered. The other communication features, described next in this section, are anchored in textual comments. In a sense, comments are the medium for all communication that happens on the site.

#### **5.1.1 User Identity**

Another crucial aspect of community oriented web sites is user identity. In order for a community to evolve over time, people need to be able to interact with each other with a minimum of persistent identity so that they may recognize each other and build up on previous interactions. On Many Eyes, a user’s identity is directly related to their activity history. By registering to enter the site, users create persistent handles that become part of their identity. Each registered user has a page that lists all of their contributions to the site: uploaded data sets, created visualizations, and comments. The page serves two purposes: it allows users to keep track of their activity in a single place and, at the same time, the accumulated history functions as an identity marker on Many Eyes. By looking at another user’s page, one can quickly get a sense of their interests.

One of the challenging aspects of sharing insights in the context of asynchronous, interactive visualizations is establishing common ground [6]. Different users need to be able to point out specific items of interest. Many Eyes supports common ground creation with two features: visualization bookmarks and visualization annotation. Whereas bookmarks allow users to capture the state of a visualization, annotation enables users to highlight specific items within a particular state of a visualization.

#### **5.1.2 Visualization annotations and bookmarks**

An interactive visualization may have hundreds of states and, a lot of times, when users wish to talk about points of interest, they may want to refer to a specific view of that visualization—defined by the settings of filtering, navigation, and parameters of visual encoding. Thus, capturing state information is essential for communication in an environment like Many Eyes. To this end we utilize a simple URL bookmarking mechanism that points back to particular views of the visualization. This approach to state sharing is common in other systems as well [12].

Every time a user creates a visualization on Many Eyes, its default view becomes a new bookmark in the system. Additionally, users have the option of including a “snapshot” of the visualization state every time they contribute a comment. Each snapshot is a new, unique URL that captures the state of the visualization at the time the comment was made. This allows users to both easily link to different views on a visualization from their comments as well as easy outside linking to visualizations on Many Eyes.

A lot of times, however, users may also need to highlight specific items within a given state of a visualization—i.e. within a given bookmark. Many Eyes supports this activity by allowing users to include graphical annotations in the comments they make.

Annotations take various forms in different visualizations types—for instance, a selection in a scatterplot looks different from a selection in a stacked graph. At the same time, it is important for visualizations across the site to share, as much as possible, a consistent visual language. In building this shared visual language, we have carefully controlled elements such as color—all visualizations share the same color palette—typography, and animated transitions. Item selection is another area where Many Eyes keeps consistency through color and active reuse of simple highlighting mechanisms across visualizations. Users are allowed to make multiple item selections (using either the “shift” or the “control” keys on the keyboard) in every visualization. We used a common highlighting color in all visualizations with the exception of the piechart, in which case selected slices are detached from the chart.

In some cases, the highlighting capabilities on Many Eyes serve additional purposes. For example, in the pie and bubble charts, selecting multiple items enables users to find the total sum of values of all selected items as well as the percentage represented by this group of points. On the network diagram, in addition to highlighting items for discussion, selection helps clarify structural details of the graph. Because highlighting a node also highlights its edges, it becomes easier to grasp the neighborhood structure of a node that otherwise might be obscured by other elements in the graph.

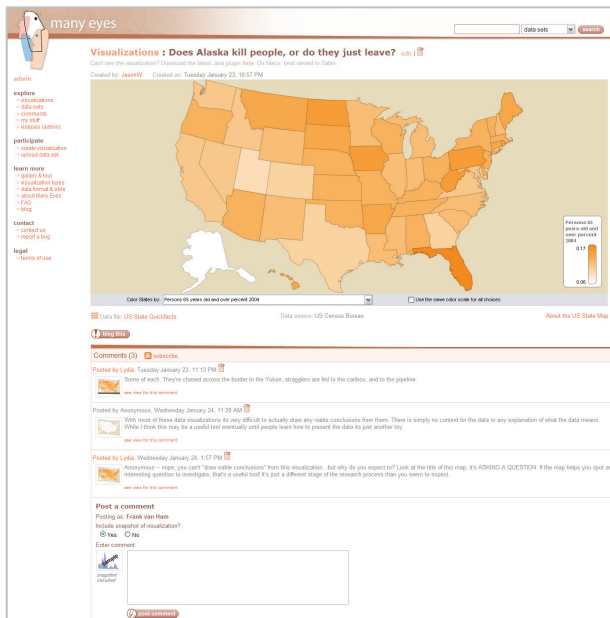


Fig. 4. The visualization and commenting interface on Many Eyes: A user has created a visualization, on which 3 other people have commented. Left of each comment is a small bookmark thumbnail that indicates the state of the user’s visualization when the comment was posted. Clicking a thumbnail modifies the state of the ‘live’ visualization to match the bookmark.

## 5.2 Off-site communication features:

One of the essential differences between Many Eyes and other collaborative visualization systems is that it exists in the context of the open web. To foster collaboration around visualizations we wanted to tap into the often lively discussions that are carried on in various blogs and other online communities. We took special care in providing easy access from outside sources to ManyEyes. Each bookmark in a visualization is published at its own unique URL, making it extremely simple for an outside source to directly link to a view on a visualization. Another feature that was targeted specifically at bloggers is a “blog this” button that appears underneath each visualization on the site. This button generates a snippet of html code that users can copy and paste onto their blogs or into the comments section. The code contains both a static snapshot of the visualization and a link back to the URL of the visualization on the Many Eyes site (see Fig. 4). This feature enables bloggers to easily embed images of visualizations into their sites and thus carry on discussions about items of interest outside of Many Eyes.

Many Eyes also features RSS feeds for data sets, visualizations, and comments. Users can sign up for feeds of the latest visualizations or data sets contributed to the site overall or they can sign up to watch for new comments to specific visualizations and data sets.

## 5.3 Related Work—visual collaboration

The concept of making visual data analysis more of a social effort is gaining popularity. Applications such as Google Earth [11] have enabled not only the creation of a wealth of user-generated geographic visualizations, but also the emergence of large numbers of discussions around visual data. An official Google Earth community has sprung up, where users discuss topics ranging from the satellite images found in the application to the availability of cartographic information about places around the world.

Although not a tool for data analysis per se, Flickr, an online service for photo sharing [10], has become the embodiment of social annotation and discussion around visual information. The site has been highly successful in enabling users to easily create discussion groups around sets of photographs and to collectively annotate the images.

As discussed above, Swivel and Data360 allow users to upload, share and discuss their own datasets with other users. Dataplace [9] is a similar site allowing users to obtain and visualize basic populations statistics on different areas in the US, but does not allow users to upload their own data. The main difference between Many Eyes and these products is that, instead of providing static business graphics, Many Eyes offers a number of interactive visualizations of user’s data. This interactivity allows users to drill down into details, view the data from different perspectives and generally makes the visualizations fun to use. The importance of the latter should not be overlooked by a site that targets the average internet user.

In terms of interactivity, Spotfire [21] and, to a lesser extent, Devise [14] also offer interactive visualizations that can be shared among users of the same application. However, Many Eyes lives on the web, where the potential audience for a visualization is greater by multiple orders of magnitude and visualizations can be linked into any online document using hyperlinks. We think that the combination of the enormous amount of collaboration infrastructure available on the web (think of blogs, forums, wiki’s and RSS feeds for example) and a webservice like Many Eyes where users can upload, visualize and share their own data brings opens new doors for communication centered visualization.

## 6 EVALUATION & EARLY USAGE

How have our design decisions worked out in practice? This section provides a short overview of activity on the site, although it is hard to capture the full range of activity.

In the first two months of the site’s life it has received about 400,000 non-robot page views, divided into 100,000 user sessions, and has gathered 1463 registered users. Users have uploaded roughly 2,100 data sets, created 1,700 visualizations, and made about 450 comments. Of the comments, about 90% have occurred on visualizations rather than data sets. This latter fact may indicate that the visualizations do have a catalyzing effect on conversation, especially given that there are more data sets than visualizations.

All of the visualization techniques have been used at least 25 times. The relative proportions (excluding visualizations created by members of our lab) are shown in . It is interesting to note that the top four visualization types are the non-standard ones. It is unclear whether this indicates an appetite for complex, experimental visualizations, or simply that people who wish to make bar charts have other options.

Table 2 Usage Statistics for the Different Visualization Types.

Visualization Component	Percentage of use
Bubble chart	15%
Network Diagram	12
Tag Cloud	11 (on site for only one month)
Treemaps	10
Bar Chart	9
Line Graph	9
World Map	8
Scatterplot	7
US State Map	7
Stack Graph for Categories	4
Block Histogram	4
Stack Graph	3
Pie Chart	1

Over 42% of registered Many Eyes users have uploaded at least one data set and 29% have created at least one visualization. Of those that uploaded data sets, 63% provided a source for the data and 40% also provided an URL for the data source. This level of data referencing is shockingly high considering that users are not required to provide sources for the data they contribute to the site.

One of the most distinctive aspects of Many Eyes is that it exists as part of the web ecosystem. In fact, we think the Internet has two

distinct characteristics that make it uniquely suited as a platform for discussion and discovery. Firstly, its massive scale means that there is always another person out there that shares your interests. This makes it easier to attain the critical mass needed for a discussion site. Secondly, the ability to easily link different information together avoids this mass being fragmented over disconnected sites and allows users to relatively easily adapt different types of tools for their personal use. As an example exemplifying both of these properties, one particular user created a network visualization that showed different textual co-occurrences of names in the bible (see Fig. 2) and linked to it on their Bible Sociometrics blog. This blogpost was subsequently picked up by different feed aggregators and received a highly ranked position. This prompted many more users to visit the original visualization on our site. A number of these users also interested in bible metrics then started uploading their own bible related datasets, and used these to create new visualizations, many of which were posted on different bible related blogs.

Our registered users range from scientists to mid level managers and self-proclaimed data geeks to sports fans. 625 of these have personally uploaded data, 425 have created a visualization and 113 have left comments on Many Eyes. Some of these visualizations quickly identified incorrect data points, even in datasets that came from respected government institutions.

Users have been in touch with us with a series of requests for new features. Visualization creators, for instance, would like to have a wider variety of maps on the site while bloggers would like to be able to embed interactive visualizations on their blogs. Visualization builders would like to add new visualization techniques to the site. Overall, feedback about Many Eyes has been positive and the variety of visualization applications—from playful gaming to serious data analysis—seems to attest to the value of the site to different users.

## 7 CONCLUSION AND FUTURE WORK

We have described the Many Eyes web site, which provides a set of visualization creation and publishing tools to a large potential audience. The architecture of a site that aims to be useable by millions is nontrivial, and we have discussed the many choices and tradeoffs in the current design. In some cases these design choices involve simplifications to or reordering of the standard visualization pipeline—allowing data transformation in the context of creating a visualization, for instance. We also have described how flexible collaborative capabilities are woven into all of the visualization components, as in our selection and bookmarking model. Finally, we have described how the site exists as part of a social ecosystem, with discussions on blogs providing a significant amount of attention and activity surrounding Many Eyes.

Future work will focus on three main areas. First, the site could benefit from a stronger set of community tools. In particular, as the collection of data sets grows, organization and quality control will become increasingly important. It would be beneficial to have mechanisms for the user community to do some of the work in organizing and filtering.

There are also natural extensions to the data model of Many Eyes. Some elements are logistical: the site would clearly gain from tools that allowed easy export from other site. More broadly, the capability of reading data from external sources would open up the possibility of “live” visualizations and the construction of composite visualization applications. All of these possibilities raise interesting questions around versioning and collaboration.

A third natural future direction is to augment the visualizations themselves. The current annotation/pointing scheme is extremely simple, and could be extended in a number of directions. For example, the site could exploit the fact that the annotations are tied to the data to allow inter-visualization brushing and selection. More generally, it would be natural to experiment with other types of visual metadata. For instance, as in [29] one could provide visual indications to show which elements of a data set of have been examined closely.

Finally, it may make sense to use Many Eyes as a platform for rapid user testing of new visualization techniques. Conceivably the site could offer an API so that third-party developers and researchers could test their own offerings. Tests could consist of simply putting new components up and watching whether they are used, or could use more sophisticated methods. For instance, in the graph visualization component users may rearrange the graph layout—and they frequently do. Would it make sense to look at these human-created layouts to deduce implicit aesthetic criteria? Such approaches seem promising, and point to the research benefits of a broadly available visualization site with a large user base.

## REFERENCES

- [1] Adai, A. et al., “LGL: creating a map of protein function with an algorithm for visualizing very large biological networks”, *J Mol Biol.* Vol 340(1), pp. 179-90, 2004.
- [2] Anupam, V., Bajaj, V., Schikore,D., Schikore, M. Distributed and collaborative visualization. *Computer*, vol. 27, no. 7, pp. 37-43 1994.
- [3] Bridgman, S and Tamassia R. “The Graph Drawing Server”, *Graph Drawing 2001*, Springer LNCS 2265, pp 44, 2002.
- [4] Brodli, K.W., Duce, D.A., Gallop, J.R., Walton, J.P., & Wood, J.D. “Distributed and Collaborative Visualization”, Blackwell Pub., 1981.
- [5] Chartall, <http://www.chartall.com>, retrieved 03-30-2007.
- [6] Clark, H.H. and Brennan, S.E. “Grounding in Communication”. In: *Perspectives on socially shared cognition*, American Psychological Association, 1991.
- [7] DabbleDB, <http://www.dabbledb.com>, retrieved 03-30-2007.
- [8] Data360, <http://www.data360.org>, retrieved 03-30-2007
- [9] DataPlace, <http://www.dataplace.org>, retrieved 03-30-2007
- [10] Flickr, <http://www.flickr.com>, retrieved 03-30-2007.
- [11] Google Earth, <http://earth.google.com>, retrieved 03-30-2007.
- [12] Heer, J., Viégas, F.B., and Wattenberg, M. “Voyagers and Voyeurs: Supporting Asynchronous Collaborative Information Visualization”, In *Proc. of CHI*, 2007.
- [13] Hill, W.C. and Hollan, J.D. “Deixis and the Future of Visualization Excellence”, In *Proc. of IEEE Visualization*, 1991.
- [14] Livny M. et al, “DEVise: integrated querying and visual exploration of large datasets”, In *Proc. ACM SIGMOD’97*, pp 301-312, 1997.
- [15] Mackinlay, J. “Automating the Design of Graphical Presentations of Relational Information”, *ACM Transactions on Graphics*, Vol. 5 ,No. 2 pp 110 – 141, 1986.
- [16] New York Times: Faces of the Dead. [http://www.nytimes.com/ref/us/20061228\\_3000FACES\\_TAB1.html](http://www.nytimes.com/ref/us/20061228_3000FACES_TAB1.html), retrieved 03-30-2007.
- [17] North, C. et al, “Visualization schemas and a web-based architecture for custom multiple-view visualization of multiple-table datasets”, *Information Visualization*, Vol 1, pp 211-229, 2002.
- [18] Quickbase, <http://www.quickbase.com>, retrieved 03-30-2007.
- [19] Roth S. et al., “Towards an Information Visualization Workspace : Combining Multiple Means of Expression”, *Human-Computer Interaction Journal*, Vol. 12 (1&2), pp 131 – 185, 1997.
- [20] Roth,S. “Visualization as a Medium for Capturing and Sharing Thoughts,” Capstone in *Proc IEEE InfoVis 2004*, p. xiii, 2004.
- [21] Spotfire, <http://www.spotfire.com>, retrieved 03-30-2007.
- [22] Swivel, <http://www.swivel.com>, retrieved 03-30-2007.
- [23] Tableau, <http://www.tableausoftware.com>, retrieved 03-30-2007.
- [24] Tang, D et al., “Design Choices when Architecting Visualizations”, *Proc IEEE InfoVis’03*, pp 41-48, 2003.
- [25] TheyRule, <http://www.theyrule.net>, retrieved 03-30-2007.
- [26] Viégas, F., boyd, d., Nguyen, D., Potter, J. & Donath, J. *Digital Artifacts for Remembering and Storytelling: PostHistory and Social Network Fragments*. In *Proc. of HICSS-37*, 2004.
- [27] Viégas, F. and Wattenberg, M. “Communication-Minded Visualization: A Call to Action”, *IBM Systems Journal*. 45(4), 2006.
- [28] Wattenberg, M. “Baby Names, Visualization, and Social Data Analysis”, In *Proc .IEEE InfoVis’05*, pp 1-7, 2005.
- [29] Wattenberg, M. and Kriss J., “Designing for Social Data Analysis”, *IEEE TVCG*. 12(4), pp 549–557, 2006.
- [30] YouTube, <http://www.youtube.com>, retrieved 03-30-2007.

# VisLink: Revealing Relationships Amongst Visualizations

Christopher Collins and Sheelagh Carpendale

**Abstract**— We present VisLink, a method by which visualizations and the relationships between them can be interactively explored. VisLink readily generalizes to support multiple visualizations, empowers inter-representational queries, and enables the reuse of the spatial variables, thus supporting efficient information encoding and providing for powerful visualization bridging. Our approach uses multiple 2D layouts, drawing each one in its own plane. These planes can then be placed and re-positioned in 3D space: side by side, in parallel, or in chosen placements that provide favoured views. Relationships, connections, and patterns between visualizations can be revealed and explored using a variety of interaction techniques including spreading activation and search filters.

**Index Terms**—Graph visualization, node-link diagrams, structural comparison, hierarchies, 3D visualization, edge aggregation.

## 1 INTRODUCTION

As information visualizations continue to play a more frequent role in information analysis, the complexity of the queries for which we would like visual explanations also continues to grow. While creating visualizations of multi-variate data is a familiar challenge, the visual portrayal of two sets of relationships, one primary and one secondary, within a given visualization is relatively new (*e.g.*, [6, 10, 17]). With VisLink, we extend this direction, making it possible to reveal relationships, patterns, and connections between two or more primary visualizations. VisLink enables reuse of the spatial visual variable, thus supporting efficient information encoding and providing for powerful visualization bridging which in turn allows inter-visualization queries. For example, consider a linguistic question such as whether the formal hierarchical structure as expressed through the IS-A relationships in WordNet [16] is reflected by actual semantic similarity from usage statistics. This is best answered by propagating relationships between two visualizations: one a hierarchical view of WordNet IS-A relationships and the other a node clustering graph of semantic similarity relationships. Patterns within the inter-visualization relationships will reveal the similarities and differences in the two views of lexical organization.

VisLink supports the display of multiple 2D visualizations, each with its own use of spatial organization and each placed on its own interactive plane. These planes can be positioned and re-positioned supporting inter-visualization comparisons; however, it is VisLink's capability for displaying inter-representational queries that is our main contribution. Propagating edges between visualizations can reveal patterns by taking advantage of the spatial structure of both visualizations. In this paper we will explain our new visualization technique in comparison to existing multi-relationship visualizations.

## 2 FORMALIZING VISUALIZATIONS OF MULTIPLE RELATIONS

VisLink extends existing approaches to visualizing multiple relationships by revealing relationships amongst visualizations while maintaining the 'spatial rights' of each individual relationship type. In order to discuss more precisely the distinctions between previous work and our contribution, we will first introduce some notation for describing multiple view visualizations.

Given a data set,  $D_A$ , and a set of relationships,  $R_A$ , on  $D_A$ , we will write this as  $R_A(D_A)$ . Note that with the relation  $R_A$  we are not refer-

ring to a strict mathematical function, but rather any relation upon a data set, for example, a type of edge among nodes in a general graph. A second set of relationships on the same data set would be  $R_B(D_A)$ , while the same set of relationships on a different but parallel data set would be  $R_A(D_B)$ . For example, if the data set  $D_A$  was housing information in Montreal, an example of  $R_A$  could be the specific house to property tax relation  $R_A(D_A)$  and a different relationship  $R_B$  could be the house size as related to the distance from transit routes  $R_B(D_A)$ . Then an example  $R_A(D_B)$  would be property tax on houses in Toronto. Creating a first visualization,  $Vis_A$ , of these relationships  $R_A(D_A)$  we will write  $Vis_A \rightarrow R_A(D_A)$  (for example, a geographic map with houses coloured based on their property tax). A second visualization,  $Vis_B$ , of the same set of relationships would be  $Vis_B \rightarrow R_A(D_A)$  (for example, a histogram of number of houses in each property tax range).

In the remainder of this section, we use this notation to define, compare, and contrast each of the current approaches to relating visualizations. We will show how VisLink provides capability beyond what is currently available.

### 2.1 Individual Visualizations

As a viewer of any given set of visualizations it is possible to do the cognitive work of developing cross visualization comparisons. For instance, visualizations can be printed and one can, by hand with pen and pencil, create annotations and/or new visualizations to develop the comparisons needed for the current task. Any relations on any data may be compared manually in this way (see Figure 1A).

### 2.2 Coordinated Multiple Views

Coordinated views provide several usually juxtaposed or tiled views of visualizations that are designed to be of use in relationship to each other (*e.g.*, Snap-Together Visualization [18]). These can be of various flavours such as  $Vis_A$ ,  $Vis_B$  and  $Vis_C$  of  $R_A(D_A)$  or perhaps  $Vis_A$  of  $R_A(D_A)$ ,  $R_B(D_A)$  and  $R_C(D_A)$ . The important factor for this visualization comparison discussion is that these coordinated views can be algorithmically linked such that actions and highlights in one view can be reflected on other views. Coordinated views allow for reuse of the spatial visual variable, thus each relationship type is afforded spatial rights. The temporarily activated visual connections can be a great advantage over finding the related data items manually but the relationships themselves are not explicitly visualized (see Figure 1B).

### 2.3 Compound Graph Visualizations

There are now a few examples of compound graph visualizations, such as overlays on Treemaps [6], ArcTrees [17], and Hierarchical Edge Bundles [10]. Figure 1C shows a simple diagram of this. Compound graph visualizations are created as follows:

**Given:** Data set  $D_A$ , containing two (or more) types of relationship:  $R_A(D_A)$ ,  $R_B(D_A)$ ,  $\dots$ ,  $R_N(D_A)$ .

**Problem:** Show multiple relationship types on the same visualization.

- Christopher Collins is a PhD Candidate with the Computer Science Department at the University of Toronto, E-mail: ccollins@cs.utoronto.ca.
- Sheelagh Carpendale is a Professor with the Computer Science Department at the University of Calgary, E-mail: sheelagh@cpsc.ucalgary.ca.

Manuscript received 31 March 2007; accepted 1 August 2007; posted online 27 October 2007.

For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.



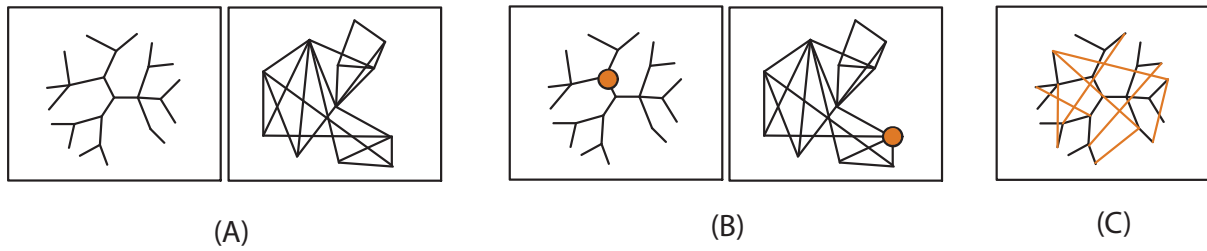


Fig. 1. Current approaches to comparing visualizations include (A) manual comparison (printed diagrams or separate programs), (B) coordinated multiple views (linked views with highlighting), and (C) compound graphs (layout based on one relationship, other relationships drawn upon it).

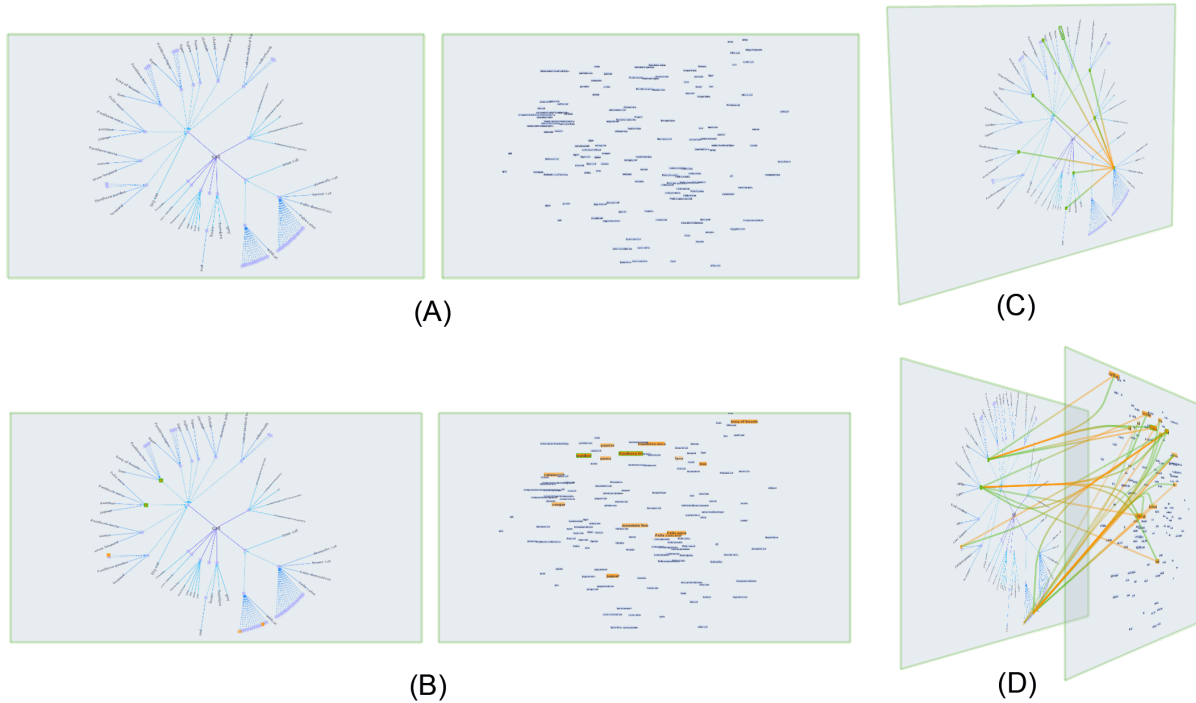


Fig. 2. VisLink encompasses existing multiple views techniques of (A) manual comparison, (B) coordinated multiple views, and (C) compound graphs. VisLink extends this continuum to direct linking of any number of multiple views (D).

**Step 1:** Choose a relationship type, *e.g.*,  $R_A$ , to be the primary relationship.

**Step 2:** Create a visualization  $Vis_A \rightarrow R_A(D_A)$ , providing an appropriate spatial layout. Since spatial organization is such a powerful factor in comprehending the given relationships, we refer to this as giving  $R_A$  ‘spatial rights’.

**Step 3:** Create a visualization of  $R_B(D_A)$  (and any other desired secondary relations) atop  $Vis_A \rightarrow R_A(D_A)$ .

This in effect creates  $Vis_A \rightarrow R_A, R_B(D_A)$  using the spatial organization of  $Vis_A \rightarrow R_A(D_A)$ . While this is an exciting step forward in comparative visualization, note that  $R_B(D_A)$  has no spatial rights of its own. That is, while viewing how the relationships in  $R_B(D_A)$  relate to  $R_A(D_A)$  is possible, there is no access to a visualization  $Vis_B \rightarrow R_B(D_A)$ . Hierarchical Edge Bundles [10] started an interesting exploration into using the spatial organization of  $R_A(D_A)$  to affect the readability of the drawing of  $R_B(D_A)$  atop  $Vis_A \rightarrow R_A(D_A)$  and also indicated possibilities of addressing the readability needs of  $R_B(D_A)$  by altering the spatial drawing of  $Vis_A \rightarrow R_A(D_A)$  so that  $R_B(D_A)$  and  $R_A(D_A)$  occupy different spatial areas. This gives  $R_B(D_A)$  partial spa-

tial rights in that its presence affects the  $Vis_A \rightarrow R_A(D_A)$  layout.

## 2.4 Semantic Substrates Visualizations

Shneiderman and Aris [20] introduce Semantic Substrates, a visualization that is both quite different and quite similar in concept to VisLink. We will use our notation to help specify this:

**Given:** Data set  $D_A$  and a set of primary relationships  $R_A(D_A)$ .

**Problem:** A given unified visualization creates too complex a graph for reasonable reading of the visualization.

**Step 1:** Partition the data set  $D_A$  into semantically interesting subsets,  $D_{A_1}, D_{A_2}, \dots, D_{A_n}$ .

**Step 2:** Use the same visualization  $Vis_A$ , with spatial rights, to create visualizations of the subsets  $Vis_A \rightarrow R_A(D_{A_1}), Vis_A \rightarrow R_A(D_{A_2}), \dots, Vis_A \rightarrow R_A(D_{A_n})$ .

**Step 3:** Juxtapose one or more of  $Vis_A \rightarrow R_A(D_{A_1}), Vis_A \rightarrow R_A(D_{A_2}), \dots, Vis_A \rightarrow R_A(D_{A_n})$ , aligned in a plane.

**Step 4:** Draw edges of  $R_A(D_A)$  across  $Vis_A \rightarrow R_A(D_{A_1}), Vis_A \rightarrow R_A(D_{A_2}), \dots, Vis_A \rightarrow R_A(D_{A_n})$  to create  $Vis_A \rightarrow R_A(D_A)$ .

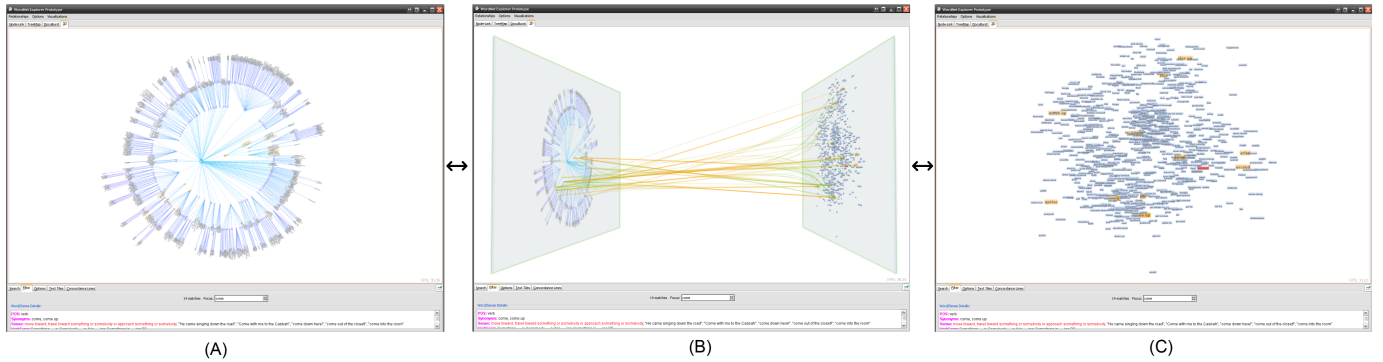


Fig. 3. Viewing modes. (A) 2D equivalency view of plane one, showing hyponyms of verb ‘move’, with highlighted search results for ‘come’. (B) Search results on plane one activate inter-plane edges, visible in 3D mode. Nodes connected to search results are highlighted on plane two, a similarity clustering of words related to ‘move’. Propagated results are also visible when plane two is viewed in 2D equivalency mode (C).

## 2.5 VisLink Visualizations

Now we will use our notation to clarify the contribution of the VisLink visualization:

**Given:** Data set  $D_A$  and a set of primary relationships  $R_A(D_A)$ ,  $R_B(D_A)$ ,  $\dots$ ,  $R_N(D_A)$ .

**Problem:** Provide a visualization that aids in improving the understanding of  $R_A(D_A)$ ,  $R_B(D_A)$ ,  $\dots$ ,  $R_N(D_A)$  by indicating how one set of relationships is related to the structure in another.

**Step 1:** Create visualizations  $Vis_A \rightarrow R_A(D_A)$ ,  $Vis_B \rightarrow R_B(D_A)$ ,  $\dots$ ,  $Vis_N \rightarrow R_N(D_A)$ , each with full spatial rights for any of  $R_A(D_A)$ ,  $R_B(D_A)$ ,  $\dots$ ,  $R_N(D_A)$  that are of interest.

**Step 2:** Place selected visualizations  $Vis_A \rightarrow R_A(D_A)$ ,  $Vis_B \rightarrow R_B(D_A)$ ,  $\dots$ ,  $Vis_N \rightarrow R_N(D_A)$  on individual planes to support varying types of juxtaposition between visualizations (at this point we are limiting these to 2D representations).

**Step 3:** Draw edges of second order relations  $T(R_A, R_B, \dots, R_N(D_A))$ , from  $Vis_i \rightarrow R_i(D_A)$  to  $Vis_{(i+1)} \rightarrow R_{(i+1)}(D_A)$  and  $Vis_{(i-1)} \rightarrow R_{(i-1)}(D_A)$  to create VisLink inter-plane edges between neighbouring planes.

So, where Semantic Substrates operates with a single visualization type and single relation across multiple subsets of a data set, VisLink can operate on multiple visualization types and multiple relationship types on a single dataset. A natural extension of VisLink is to inferred or indirect relations across multiple data sets:

**Given:** Data sets  $D_A$ ,  $D_B$ ,  $\dots$ ,  $D_N$  and the existence meaningful relationships,  $T(D_i, D_j)$ , among datasets such that  $(i, j)$  are any of  $A$ ,  $B$ ,  $\dots$ ,  $N$ .

**Visualize:** VisLink can be used with no further extensions to relate  $Vis_A \rightarrow R_A(D_A)$ ,  $Vis_B \rightarrow R_B(D_B)$ ,  $\dots$ ,  $Vis_N \rightarrow R_N(D_N)$ , by using  $T(D_i, D_j)$  to create inter-plane edges. An example of cross-dataset visualization is presented in Section 5.

We have presented a series of multi-relation visualizations, differing in the level of visual and algorithmic integration between relations and the amount of spatial rights accorded to secondary relations. VisLink can be used equivalently to any of the mentioned multi-relation visualization approaches (see Figure 2A–C) and extends the series to simultaneously provide equal spatial rights to all relations for which a visualization can be created, along with close visual and algorithmic integration of different relations (see Figure 2D).

## 3 VISLINK: COMPARISON WITH VISUALIZATION PLANES

In order to provide for a visualization space in which multiple data-related visualizations can be analyzed, we have developed VisLink. We start our explanation with a very brief description of the lexical data set and the lexical data relationships which are used to illustrate VisLink’s functionality and interactive capabilities. Next we show a sample set of 2D lexical visualizations displayed on visualization planes within VisLink, followed by the possible interactions with these

visualization planes. Then the inter-visualization edges are explained and the ability to use inter-plane edge propagation to answer complex queries is presented.

### 3.1 Visualizations of Lexical Data

The example figures in this paper are drawn from application of VisLink to a lexical data set. This is an area of interest to computational linguists, and several visualizations using lexical data have been reported (e.g., [5, 13]).

Using our formalism, we have a dataset  $D_A$  containing all the words in the English language. There are many types of relationships among words, for example, the lexical database WordNet [16] describes the hierarchical IS–A relation over synsets, which are sets of synonymous words. For example,  $\{lawyer, attorney\}$  IS–A  $\{occupation, job\}$ . The IS–A relation is also called hyponymy, so *chair* is a hyponym of *furniture*. We use hyponymy to build animated radial graphs [22], which serve as our  $Vis_A \rightarrow R_A(D_A)$ . Synsets are shown in the radial graph as small squares, and the synonymous words that make up the set are shown as attached, labelled, nodes. An example 2D radial hyponymy graph is in Figure 3A.

Words can also be related by their similarity. Similarity can be a surface feature, for example, orthographic (alphabetic) similarity, or it can be based on underlying semantics. We use a force-directed layout [1] to perform similarity clustering on words. In our examples we use orthographic similarity, so that all words are connected to all others by springs whose tension coefficient is inversely related to number of consecutive character matches in the substring, starting at the beginning. Words that start with the same letters will cluster together. This is a very different structure than hyponymy and serves as  $Vis_B \rightarrow R_B(D_A)$ . An example 2D alphabetic clustering visualization is in Figure 3C. We have also experimented with clustering using the semantic similarity measures implemented by Pedersen *et al.* [19], for example similarity as measured by lexical overlap in the dictionary definitions of words. However, those measures did not produce visible clusters and further investigation is needed into the appropriate relationship between the similarity measure and the spring coefficient.

Using VisLink, we investigate relations between the hyponymy layout of synsets and the orthographic clustering layout of words. With this, we can investigate questions such as: do some synsets contain high concentrations of orthographically similar words?

Data is loaded into the VisLink lexical visualization by looking up a synset in WordNet to root the hyponymy tree. The orthographic clustering is then populated with the relevant words from the dataset.

### 3.2 Navigation and Plane Interaction

VisLink is a 3D space within which any number of 2D semi-transparent visualization planes are positioned. These visualization planes act as virtual displays, upon which any data visualization can

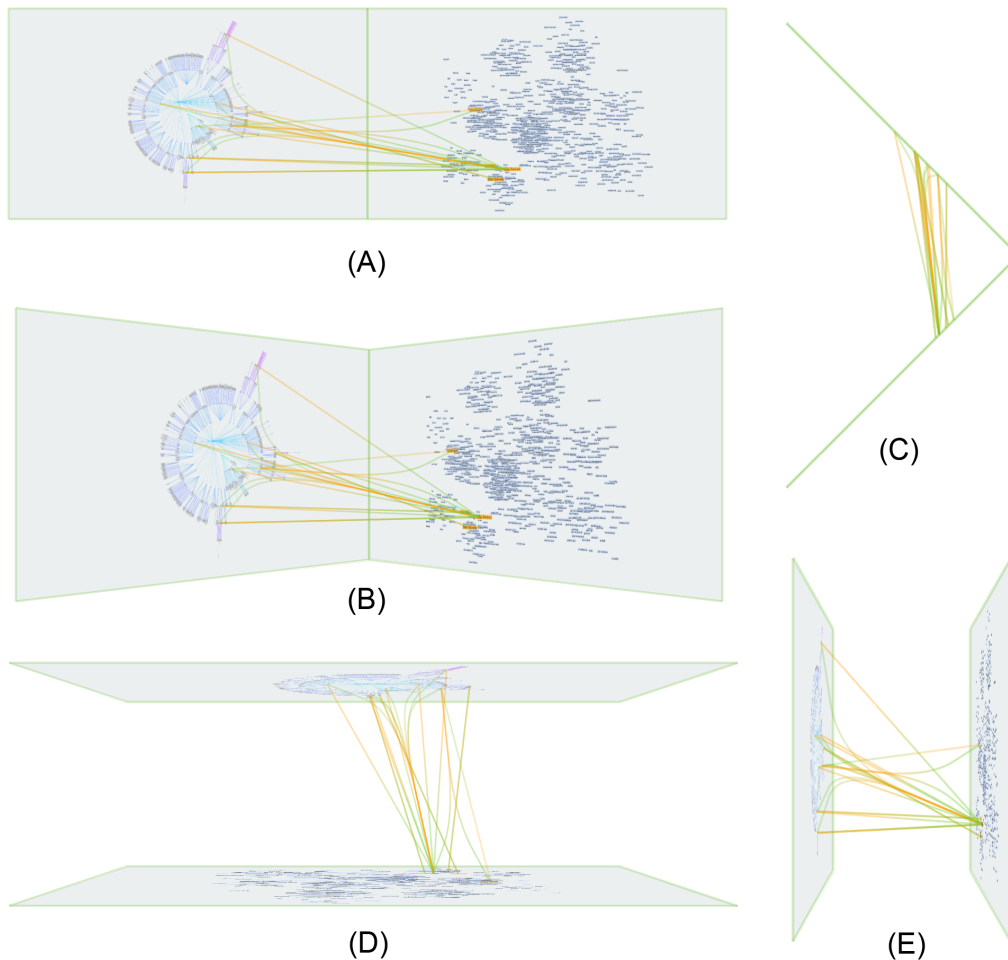


Fig. 4. Keyboard shortcuts provide for animated transition to default views, easing navigation in the 3D space. Views are (A) flat, (B) book, (C) book top, (D) top, and (E) side.

be drawn and manipulated. They can be rotated and shown side by side similar to multi-program or coordinated views, or rotated in opposition with included connections. Interaction and representation with each plane remains unchanged (representations do not relinquish any ‘spatial rights’ nor any ‘interaction rights’).

While VisLink is a 3D space, the visualization planes are 2D equivalents of a display, similar to windows in Miramar [14] or view-ports in the Web Forager [3]. We provide view animation shortcuts to transition between 2D and 3D views. Similar to interaction provided by Miramar, any visualization plane may be selected, activating an animated transition in which the selected plane flies forward and reorients to fill display space. When a plane is selected, 3D interaction widgets and inter-plane edges are deactivated, and the display becomes equivalent to 2D (see Figure 3). Because VisLink visualization planes have the same virtual dimensions as the on-screen view-port, transition between 2D plane view and 3D VisLink view does not require any resizing of the selected plane. When the plane is deselected, it falls back into the VisLink space, reverting to the original 3D view.

Interaction with the visualization on a visualization plane is always equivalent to 2D: mouse events are transformed to plane-relative coordinates and passed to the relevant visualization (irrespective of the current position and orientation of the plane). Visualizations can be manipulated directly in the 3D space (using equivalent-to-2D mode is not necessary). Thus interaction techniques developed for 2D visualizations become immediately available in VisLink. For example, we provide for a radial node-link view of the WordNet hyponymy (IS-A) relation, restricted with a generalized fish eye view to show only nodes

of distance  $N$  or less from the central focus. The focus node can be reselected by a mouse click, activating radial layout animation [22]. Double clicking any node restricts the view to the tree rooted at that node, providing for drill-down capability. Drill down and other data reload interactions are propagated to all planes. Interaction techniques such as panning and zooming in 2D are provided by clicking and dragging on a visualization plane the same as one would on an equivalent stand-alone 2D visualization.

In addition to interaction with the visualizations on VisLink planes, we also provide for interaction with the planes themselves. While the usual capabilities for navigation in a 3D space (pan, zoom, rotate of camera position) are available in VisLink, in providing a 3D perspective projection virtual space, we must address the difficulties that arise from 6-degrees-of-freedom (DOF) control with 2-DOF input devices [2]. Free navigation can result in disorientation and non-optimal viewing positions, while free manipulation of 3D objects can result in difficulty achieving precise desired positioning.

Therefore, we also provide shortcuts for cinematic animated repositioning of the camera and planes to preset viewpoints [14]. These viewpoints allow visualization planes to be viewed from the front (planes parallel and side by side) (see Figure 4A), with relative plane orientation of book view (planes perpendicular and meet at an edge) (see Figure 4B), top (see Figure 4C and D), or in opposition (planes parallel and stacked) (see Figure 4D and E). By choosing one of these viewpoints, users can recover from any disorienting manipulation.

As a solution to 2D plane interaction in a 3D space, we follow McGuffin *et al.* [15] and provide for manipulation of visualization

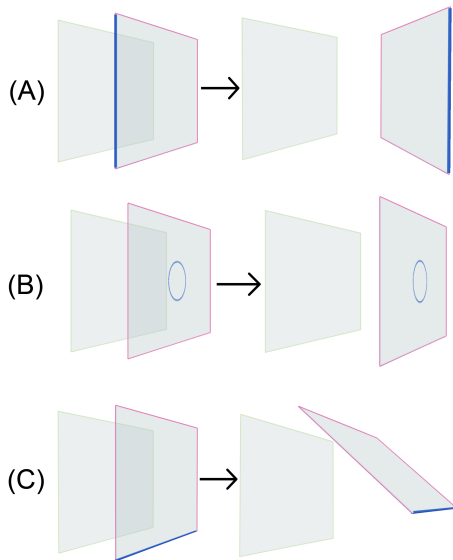


Fig. 5. Visualization planes are independently manipulated with three widgets: (A) side ‘book pages’ rotation, (B) center ‘accordion’ translation, and (C) bottom ‘garage door’ rotation.

plane position and orientation using a set of restricted movement widgets. Edge widgets provide for hinge movement (up to 90 degrees) about the opposite edge, and a center widget provides for translation, accordion style, along the axis between the planes (see Figure 5). Widgets become visible when the pointer is over their position, otherwise they are hidden from view to prevent data occlusion.

### 3.3 Adding Inter-Plane Edges

Edges are drawn in 3D to bridge adjacent visualization planes. Relationships between the visualizations can either be direct (nodes representing the same data are connected across planes) or indirect (items on different planes have relations defined within the data).

For example, in our lexical visualization, we examine the formal structure of WordNet hyponymy (the IS-A relation) on one plane, and the clustering of words based on their similarity on another. The inter-plane relationship in this case is direct: nodes on plane one represent the same data as nodes on plane two. In this case, it is the difference in the spatial organization of the layouts that is of interest. In essence, the pattern of inter-plane edges reveals a second-order relation: the relationship between different types of node relations on the same data. If the clustering by similarity approximates the formal structure, edges from synonyms in the structured data will go to the same cluster (*i.e.*, edges from synonyms will be parallel).

Indirect relations can also be visualized. For example, a visualization plane could be populated with a general graph about self-declared friendships in a social networking system. A second visualization plane could be populated with a tag cloud from a folksonomy, for example a bookmark sharing database. A third visualization plane could be populated with a visualization of the hypertext links between bookmarked pages. The three types of indirect inter-plane connections could be derived from three cross-dataset rules: PERSON used TAG, PAGE tagged with TAG, and PERSON bookmarked PAGE. With effective inter-plane edge management and data filtering, patterns between planes in such a visualization could reveal people who share tagging habits, or bookmarked pages with similar tag sets.

All inter-plane edges are specified with a single source node on plane  $i$  and one or more target nodes on plane  $j$ . Single source to single target edges are drawn as straight lines. Single source to many target edges are drawn using multiple curves calculated with corner-cutting

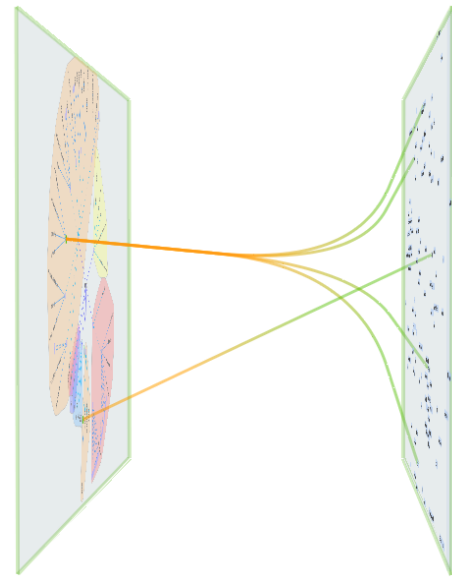


Fig. 6. VisLink inter-plane edge detail: one-to-one edges are straight, one-to-many edges are bundled. Alpha blending provides for stronger appearance of bundled edges.

[4]. For each curve from the source to a target, the starting control point is set as the source node, a middle control point is set as the average (world coordinates) position of all target nodes and the source, and the end point is set as the target. Five iterations of corner-cutting provide for smooth curves which start along the same straight line and then diverge as they approach their targets. By using alpha blending, the more semi-transparent curves that are coincident, the stronger the bundled edges appear (see Figure 6). Inter-plane edge positions are recalculated as appropriate so that edges remain fluidly attached to their source and target nodes throughout all manipulations of the constituent visualizations, plane positions, and the 3D viewpoint.

For visual clarity, edges are drawn between items on adjacent planes only. For more than two visualization planes, if the data contains relations among all visualizations, these relations can be explored by reordering the visualization planes using the center translation (accordion) widget to move planes along the inter-plane axis. As a plane passes through another, the rendering is updated to show the relations between the new neighbours. Similar to axis ordering in parallel coordinates plots [11], the ordering of visualization planes strongly effects the visibility of interesting patterns in the data. Investigation into methods for choosing plane orderings is left for future research.

### 3.4 Using Inter-Plane Edges

Inter-plane edges can be revealed either on a per-plane basis (see Figure 7) or a per-node basis (see Figure 8). Activating an entire plane can reveal structural patterns that may exist between the visualizations, while individual node activation provides for detailed views of particular relations.

We provide for spreading node activation between planes, which adds additional analytic power to VisLink. When a node is manually activated on one plane, it is highlighted in orange with a green border and all inter-plane edges originating at that node are revealed. The target nodes for those edges are then activated. Edges originating at these nodes are then drawn and the activation is propagated iteratively up to a user-selected number of ‘reflections’ between planes. Deactivation of a node reverses the process, spreading the deactivation and hiding edges. The level of activation exponentially decays with each iteration.

Nodes are assigned activation values from 0 (deactivated) to 1 (manually activated by user through selection, search, or plane acti-

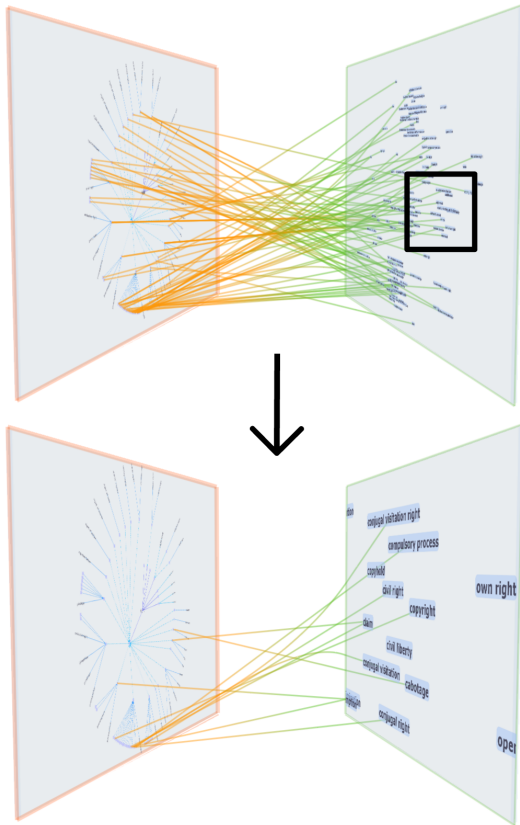


Fig. 7. The left plane is activated, revealing all edges from it. Through a click and drag on the right plane, a 2D zoom is performed, isolating a cluster of interest. The inter-plane edges are filtered in real time to show only those connecting visible nodes, revealing that this lexical cluster is related to a region of the WordNet hyponymy tree near the bottom.

vation). Node activation values determine inter-plane edge visibility: edges between nodes with non-zero activation are revealed. Level of activation is inversely related to the alpha transparency of activated nodes and the inter-plane edges. So, the more transparent an activated node or edge, the further it is from a user-selected fully-activated node. Edge colour is used to indicate the direction of spreading activation. For each edge, the third closest to the source of edge activation is orange, the middle third is interpolated from orange to green and the final third, closest to the edge target, is green. Along with edge transparency decay, edge colouration will help an analyst follow the path of spreading activation. However, tracing a series of edges across planes may be a difficult task, even with the visual support provided through colouration and transparency. We plan to investigate techniques such as animated edge propagation to help trace relationships amongst visualizations.

Inter-plane edges support cross-visualization queries. For example, alphabetic clustering, while a common organization for word search, is not useful for finding synonyms. Using VisLink to propagate an edge from a selected word in the clustered graph to a WordNet hierarchy will find this word within its synset structure, propagating back will find its synonyms within their alphabetic structure, allowing quick answers to questions such as, “Across all senses, which synonyms of ‘locomotion’ start with ‘t’?” This analysis is illustrated in Figure 8.

Inter-plane edges are only shown among visible nodes. So, if a technique such as filtering through degree-of-interest or distance measures, or clipping through zooming and panning the visualization on a plane causes some nodes to be invisible, their edges are not drawn. This can be used as an advantage for exploring the space of inter-plane edges: by filtering the view on a plane, the inter-plane edges can also be fil-

tered (see Figure 7). Conversely, search techniques can be provided to reveal and activate nodes that match a query, thereby also activating their inter-plane edges (see Figure 3).

#### 4 IMPLEMENTATION DETAILS

VisLink is implemented in Java, using the Java2D-Java OpenGL (JOGL) bridge to import any Java2D rendering onto a visualization plane. We have augmented the popular *prefuse* interactive visualization toolkit [9] with the VisualizationPlane class, which implements the same API as the default 2D *prefuse* Display, and the InterPlaneEdge class, which handles edge drawing between planes. The result is that our visualization plane can accept any *prefuse* visualization without any changes. Interaction techniques on *prefuse* visualizations are also handled equivalently. In addition to providing for easy integration of existing visualizations with VisLink, this implementation provides for efficient rendering of the 3D space, achieving frame rates greater than 30fps on standard hardware (Intel Pentium 4, 3.9GHz processor with an ATI Radeon 550 graphics card). The *prefuse* visualizations are shown on the visualization planes as textures, updated only when *prefuse* calls for a display repaint. Inter-plane edges can be specified in the data set by referencing source and target visualization plane and node indices, or can be defined by a rule, such as, “Create inter-plane edges among nodes with matching labels” (rules such as these must be translated into code that produces paired node indices). Because the *prefuse* visualizations are drawn as textures on a 2D plane, VisLink could easily be extended to draw other shapes of visualization objects, such as cubes or spheres.

#### 5 LINKING EXISTING VISUALIZATIONS

To demonstrate the ability of VisLink to add analytic power to existing *prefuse*-based visualizations, we used VisLink to bridge several of the demonstration applications that are distributed with the *prefuse* source code [9] (with minor colour changes). Data on the occupations of members of the 109th Congress before election was mined from the Congressional Directory,<sup>1</sup> along with the zip codes they represent. This was combined with databases of zip code locations and fundraising totals of candidates in three recent federal elections, both provided with the *prefuse* distribution. We used three visualization planes and defined indirect relations between them.

First, a *prefuse* Treemap [12] was used to show the relative popularities of various occupations before election (Figure 9, left). This was linked through the rule CANDIDATE had OCCUPATION to the *prefuse*-provided *congress* visualization by Heer [8]. *congress* is a scatterplot of individual fundraising success, ordered along the x-axis alphabetically by state of candidacy (Figure 9, center). This plot shows the candidates’ party through node colour and whether they were running for the House or Senate through node shape. The y-axis shows fundraising success, and the range can be interactively altered with a slider (not shown in figure). This was linked to the *prefuse* reimplementations of the *zipdecode* [7] visualization of zip code geographic locations (Figure 9, right) through the rule CANDIDATE represents ZIP CODE. Inter-plane edges link occupations to candidate nodes and candidates to map regions they now represent. Complex questions such as, “Where did the most successful fundraising former journalist get elected?” can be quickly answered. To implement this visualization, the bulk of the work came through creating and parsing the new database (occupations and zip codes) to generate inter-plane edges from our rules.

#### 6 DISCUSSION

The VisLink technique offers a new way to look at the relationships amongst visualizations, but there remain several difficulties and unresolved issues for future research. The creation of a VisLink visualization starts with the selection of the constituent visualizations to compare. Making this selection — finding appropriate data and choosing appropriate representations — is as difficult within VisLink as it is in everyday visual analytics work, and may be best handled by data and

<sup>1</sup><http://www.gpoaccess.gov/cdirectory>

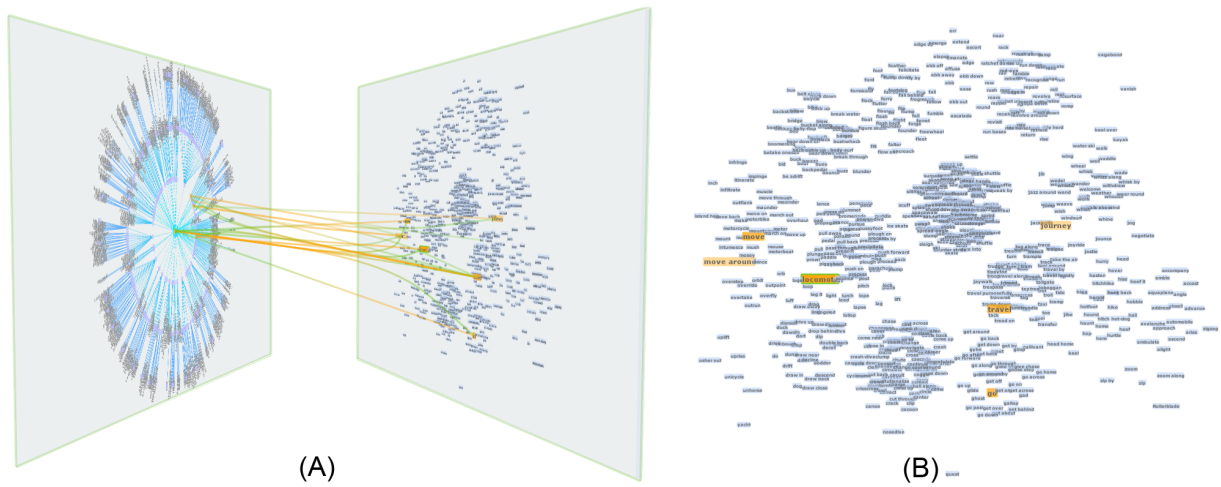


Fig. 8. Node activation and edge propagation. Nodes highlighted through spreading activation (orange without green border) reveal the alphabetic clustering of synonyms of the manually activated node ('locomotion', orange with green border), as discovered through spreading activation to the WordNet hyponymy graph.



Fig. 9. VisLink was applied to bridge existing *prefuse* visualizations. Views of the constituent visualizations, from 2D equivalency mode, are shown along the bottom. The Treemap node 'journalist' is activated, propagating inter-plane edges to the scatterplot (showing journalists are not particularly outstanding fundraisers), and onward to the zip code regions that elected journalists now represent.

visual experts. Some visualizations, such as node-link diagrams, seem to work better with inter-plane edges than others, such as Treemaps and other types of embedded hierarchy, where it is more difficult to see the connections to non-leaf nodes.

For visualizations with rich sets of inter-plane relations, the familiar spaghetti graph of edge congestion can quickly become a problem. Through bundling of edges, individual node activation, filtering techniques, and the ability to view the edge set from a series of angles, we have attempted to provide tools to handle this. However, additional techniques, for example edge lenses [21] for 3D spaces, may improve the situation. The edge bundling technique we use works only for one-to-many edge sets. Many-to-many edge bundling as reported by Holten [10] requires a hierarchical structure as an invisible backbone. In the datasets we used, such a structure was not available, but this may be a promising area for future research.

Because VisLink contains any number of visualizations which may be pre-existing, the selection of colours for inter-plane edges is challenging. The orange-to-green colour scheme was selected because it interfered the least with the existing (predominantly blue) visualizations we imported into VisLink, and worked well both against a white background (for print) and a black background (on screen). However, orange-to-green is difficult to perceive for people with some forms of colour blindness. Inter-plane edge colouring will likely have to be customized to the constituent visualizations.

When working in a 3D space, issues of perspective must be considered. It is possible that perspective projection introduces a visual bias for closer regions of the planes and closer inter-plane edges. Directional bias may be introduced by the default views (side view presents bias toward vertical inter-plane patterns). 2D false symmetry effects may also occur. An analyst must be careful to view a VisLink visualization from several directions before drawing conclusions about apparent patterns in the data.

We have described VisLink primarily with examples from a single data set. In future work, we will apply VisLink to a rich set of problems in linguistic data analysis and other areas. The techniques and prototype we have described have not yet been experimentally evaluated. A comparative study against the existing techniques for multiple relationship visualization is necessary to understand the usability and utility of VisLink in more detail. Opportunities also exist to expand the capabilities of inter-representational queries, for example, by providing for a rich query language that can filter each visualization plane separately.

## 7 CONCLUSION

In this paper we have described VisLink, a visualization environment in which one can display multiple 2D visualizations, re-position and re-organize them in 3D, and display relationships between them by propagating edges from one visualization to another. Through reuse of the powerful spatial visual variable, we have introduced a method for visualizing multiple relations without any relation relinquishing its spatial rights.

The VisLink environment allows the viewer to query a given visualization in terms of a second visualization, using the structure in the second visualization to reveal new patterns within the first. By choosing a set of data items in visualization A and doing a one level propagation to visualization B, VisLink shows where items in A are related to items in B. Propagating the edges back again reflects the information gathered from visualization B to the structure of visualization A. Thus, using the example in Figure 8, starting from a similarity-based word visualization A, propagating edges from a chosen word into WordNet visualization B and back again reveals synonyms of the selected word in visualization A. Through spreading activation, bundled edges can be propagated between visualizations to any chosen depth.

VisLink displays multiple 2D visualizations on visualization planes while maintaining full 2D interactivity for each component visualization. 3D interaction widgets are provided to simplify 3D interaction and navigation. Relationships among visualizations can be revealed using methods such as selection and filtering for addressing edge congestion. Ongoing research will investigate techniques for managing

edge congestion, such as alternative bundling techniques and the use of interaction tools to isolate edge sets of interest. In future work, through application to additional problems, and evaluation against related techniques, we will develop a clearer understanding of the usability and utility of the techniques and prototype we have described.

## ACKNOWLEDGEMENTS

Thanks to Gerald Penn, Petra Neumann, Mark Hancock, Tobias Isenberg, Uta Hinrichs, and Matthew Tobiasz for their assistance. Funding for this research was provided by NSERC, iCore, and NECTAR.

## REFERENCES

- [1] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1999.
- [2] D. A. Bowman, E. Kruijff, J. J. LaViola, Jr., and I. Poupyrev. *3D User Interfaces*. Addison-Wesley, 2005.
- [3] S. K. Card, G. G. Robertson, and W. York. The WebBook and the Web Forager: An information workspace for the world-wide web. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, 1996.
- [4] G. A. Chaikin. An algorithm for high speed curve generation. *Computer Graphics and Image Processing*, 3(12):346–349, 1974.
- [5] C. Collins. Docuburst: Radial space-filling visualization of document content. Technical Report KMDI-TR-2007-1, Knowledge Media Design Institute, University of Toronto, 2007.
- [6] J.-D. Fekete, D. Wang, N. Dang, A. Aris, and C. Plaisant. Overlaying graph links on Treemaps. In *Proc. of IEEE Symp. on Information Visualization, Poster Session*, pages 82–83, 2003.
- [7] B. Fry. zipdecode. <http://acg.media.mit.edu/people/fry/zipdecode/>, 2007.
- [8] J. Heer. congress. <http://www.prefuse.org/gallery/congress>, 2007.
- [9] J. Heer, S. K. Card, and J. A. Landay. prefuse: a toolkit for interactive information visualization. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM Press, Apr. 2005.
- [10] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics (Proc. of IEEE Symp. on Information Visualization)*, 12(5):741–748, Sept.–Oct. 2006.
- [11] A. Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proc. of IEEE Visualization*, pages 361–378, 1990.
- [12] B. Johnson and B. Shneiderman. Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In *Proc. of IEEE Visualization*, pages 284–291. IEEE Computer Society, 1991.
- [13] J. Kamps and M. Marx. Visualizing WordNet structure. In *Proc. of the 1st International Conference on Global WordNet*, pages 182–186, 2002.
- [14] J. Light and J. Miller. Miramar: A 3d workplace. In *Proc. of IEEE Intl. Professional Communication Conf.*, pages 271–282, 2002.
- [15] M. J. McGuffin, L. Tancau, and R. Balakrishnan. Using deformations for browsing volumetric data. In *Proc. of IEEE Visualization*, pages 401–408, Oct. 2003.
- [16] G. A. Miller, C. Fellbaum, R. Tengi, S. Wolff, P. Wakefield, H. Langone, and B. Haskell. WordNet: A lexical database for the English language, Mar. 2007.
- [17] P. Neumann, S. Schlechtweg, and M. S. T. Carpendale. Arctrees: Visualizing relation in hierarchical data. In K. W. Brodlie, D. J. Duke, and K. I. Joy, editors, *Proc. of Eurographics - IEEE VGTC Symp. on Visualization*, pages 53–60. The Eurographics Association, 2005.
- [18] C. North and B. Shneiderman. Snap-together visualization: A user interface for coordinating visualizations via relational schemata. In *Proc. of Advanced Visual Interfaces*, pages 128–135, May 2000.
- [19] T. Pedersen, S. Banerjee, and S. Patwardhan. Maximizing semantic relatedness to perform word sense disambiguation. Technical Report UMSI 2005/25, University of Minnesota Supercomputing Institute, 2005.
- [20] B. Shneiderman and A. Aris. Network visualization by semantic substrates. *IEEE Transactions on Visualization and Computer Graphics (Proc. of IEEE Symp. on Information Visualization)*, 12(5):733–740, Sept.–Oct. 2006.
- [21] N. Wong, S. Carpendale, and S. Greenberg. EdgeLens: An interactive method for managing edge congestion in graphs. In *Proc. of IEEE Symp. on Information Visualization*, pages 51–58, 2003.
- [22] K.-P. Yee, D. Fisher, R. Dhamija, and M. Hearst. Animated exploration of dynamic graphs with radial layout. In *Proc. of IEEE Symp. on Information Visualization*, pages 43–50, 2001.

# Graph Visualization and Navigation in Information Visualization: A Survey

Ivan Herman, *Member, IEEE Computer Society*, Guy Melançon, and M. Scott Marshall

**Abstract**—This is a survey on graph visualization and navigation techniques, as used in information visualization. Graphs appear in numerous applications such as web browsing, state-transition diagrams, and data structures. The ability to visualize and to navigate in these potentially large, abstract graphs is often a crucial part of an application. Information visualization has specific requirements, which means that this survey approaches the results of traditional graph drawing from a different perspective.

**Index Terms**—Information visualization, graph visualization, graph drawing, navigation, focus+context, fish-eye, clustering.

## 1 INTRODUCTION

ALTHOUGH the visualization of graphs is the subject of this survey, it is *not* about graph drawing in general. Excellent bibliographic surveys [4], [34], books [5], or even on-line tutorials [26] exist for graph drawing. Instead, the handling of graphs is considered with respect to information visualization.

Information visualization has become a large field and “subfields” are beginning to emerge (see, for example, Card et al. [16] for a recent collection of papers from the last decade). A simple way to determine the applicability of graph visualization is to consider the following question: *Is there an inherent relation among the data elements to be visualized?* If the answer to the question is “no,” then data elements are “unstructured” and the goal of the information visualization system might be to help discover relations among data through visual means. If, however, the answer to the question is “yes,” then the data can be represented by the nodes of a graph, with the edges representing the relations.

Information visualization research dealing with unstructured data has a distinct flavor. However, such research is *not* the subject of this survey. Instead, our discussion focuses on representations of structured data, i.e., *where graphs are the fundamental structural representation of the data*. Information visualization has specific requirements, which means that we will approach the results of traditional graph drawing from a different perspective than other surveys.

### 1.1 Typical Application Areas

Graph visualization has many areas of application. Most people have encountered a file hierarchy on a computer system. A file hierarchy can be represented as a tree (a special type of graph). It is often necessary to navigate through the file hierarchy in order to find a particular file. Anyone who has done this has probably experienced a few of the problems involved in graph visualization: “Where am

I?” “Where is the file that I’m looking for?” Other familiar types of graphs include the hierarchy illustrated in an organizational chart and taxonomies that portray the relations between species. Web site maps are another application of graphs, as well as browsing history. In biology and chemistry, graphs are applied to evolutionary trees, phylogenetic trees, molecular maps, genetic maps, biochemical pathways, and protein functions. Other areas of application include object-oriented systems (class browsers), data structures (compiler data structures in particular), real-time systems (state-transition diagrams, Petri nets), data flow diagrams, subroutine-call graphs, entity relationship diagrams (e.g., UML and database structures), semantic networks and knowledge-representation diagrams, project management (PERT diagrams), logic programming (SLD-trees), VLSI (circuit schematics), virtual reality (scene graphs), and document management systems. Note that the information isn’t always guaranteed to be in a purely hierarchical format—this necessitates techniques which can deal with more general graphs than trees.

### 1.2 Key Issues in Graph Visualization

The size of the graph to view is a key issue in graph visualization. Large graphs pose several difficult problems. If the number of elements is large, it can compromise performance or even reach the limits of the viewing platform. Even if it is possible to layout and display all the elements, the issue of viewability or usability arises because it will become impossible to discern between nodes and edges (see Fig. 1, although this tree is by no means a very complex one). In fact, usability becomes an issue even before the problem of discernability is reached. It is well-known that comprehension and detailed analysis of data in graph structures is easiest when the size of the displayed graph is small. In general, displaying an entire large graph may give an indication of the overall structure or a location within it, but makes it difficult to comprehend. These issues form the context for most of this survey.

Other than the usual reference to information overload and the occasional reference to the gestalt principle, papers in information visualization rarely apply cognitive science

• The authors are with the Centre for Mathematics and Computer Sciences, CWI, Kruislaan 413, PO Box 94079, 1090 GB Amsterdam, The Netherlands. E-mail: {I.Herman, G.Melancon, M.S. Marshall}@cwi.nl.

For information on obtaining reprints of this article, please send e-mail to: [tcvg@computer.org](mailto:tcvg@computer.org), and reference IEEECS Log Number 111225.



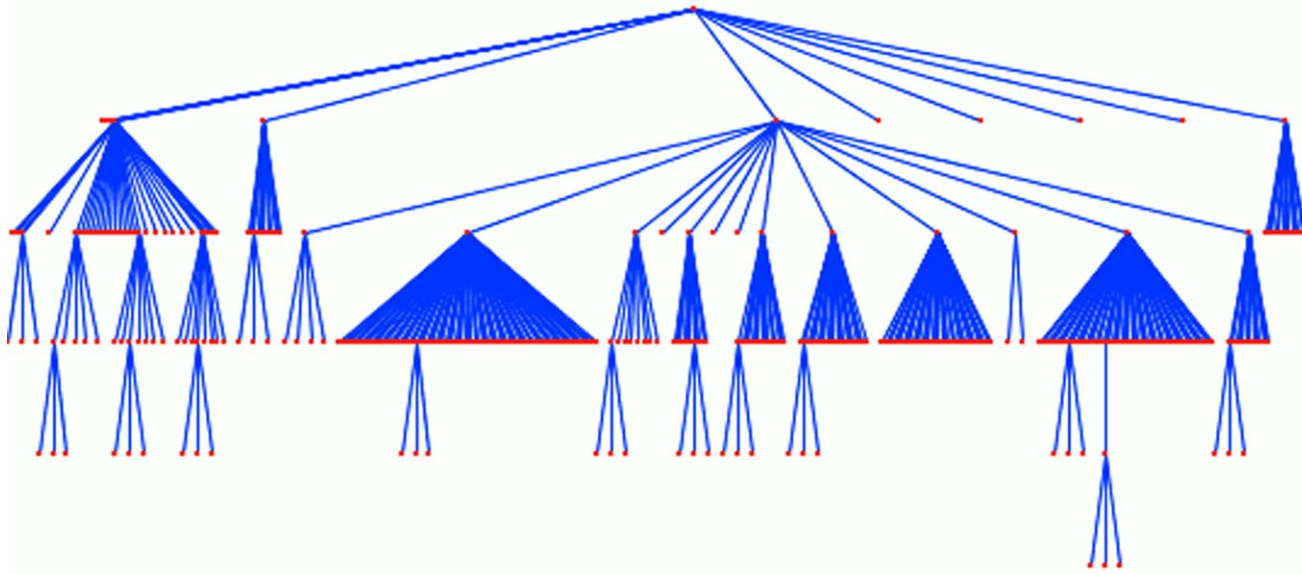


Fig. 1. A tree layout for a moderately large graph.

and human factors. This is for no lack of trying; very few of the findings in cognitive science have practical applications at this time and very few usability studies have been done. Cognitive aspects are undoubtedly a subject for future research. For this reason, an objective evaluation of the merits of a given approach is difficult. The reader has to bear this limitation in mind when various techniques are presented.<sup>1</sup>

The rest of this survey is organized as follows: In Section 2, we try to give an impression of graph layout issues and limitations with regard to scalability. Then, we discuss several approaches to navigation of large graphs (Section 3), followed by methods of reducing visual complexity through reorganization of the data (Section 4). Afterwards, we discuss a few application systems that implement many of the techniques described in this survey (Section 5). To help the reader pursue further research and development, we have listed the various sources of information that we found particularly important for graph visualization (Section 6) and provided an extensive list of references.

## 2 GRAPH LAYOUT

This section looks at the current results in graph drawing and layout algorithms, but from the point of view of graph visualization in information visualization. As we will see, this point of view differs, in many respects, from the traditional view of the Graph Drawing community. We will give an account of the available results and discuss their relevance for graph visualization, although, in general, we will not go too far into the technical details. For those

1. Ware's new book [123] may become an important source of information in this area.

desiring more information, we recommend the excellent book from Battista et al. [5] as one of the best starting points.

### 2.1 Background of Graph Drawing

The Graph Drawing community<sup>2</sup> grew around the yearly Symposia on Graph Drawing (GD 'XX conferences), which were initiated in 1992 in Rome. Springer-Verlag publishes the proceedings of the conference in the LNCS series, which contains new layout algorithms, theoretical results on their efficiency or limitations, and systems demonstrations. The recent electronic *Journal of Graph Algorithms and Applications* is dedicated to papers concerned with design and analysis of graph algorithms, as well as with experiences and applications.

The basic graph drawing problem can be put simply: Given a set of nodes with a set of edges (relations), calculate the position of the nodes and the curve to be drawn for each edge. Of course, this problem has always existed for the simple reason that a graph is often defined by its drawing. Indeed, Euler himself relied on a drawing to solve the "Königsberger Brückenproblem" in his 1736 paper (see the recent book of Jungnickel [74]). The annotated bibliography by Battista et al. [4] gathers hundreds of papers studying what a *good* drawing of a graph is. That is, where the problem becomes more intricate: It requires the definition of properties and a classification of layouts according to the type of graphs to which they can be applied. For example, a familiar property is *planarity*—whether it is possible to draw a graph on the plane with no edge crossing. Layout algorithms may be categorized with respect to the type of layout they generate. For example, grid layouts position nodes of a graph at points with integer coordinates. Other categories of layouts are defined by the methodology on which they are based. For example, nondeterministic

2. <http://www.cs.brown.edu/people/rt/gd.html>.

approaches form a category that uses algorithms such as force-directed models or simulated annealing. Each class of graphs and layouts thus generates its own set of problems. Planarity, for example, raises problems such as:

- Planarity tests for graphs: Is it possible to draw a graph without edge-crossings?
- Planar layout algorithms according to various constraints: Given that a graph is planar, find a layout satisfying a group of constraints.

Many constraints in use are also expressed in terms of *aesthetic rules* imposed on the final layout. Nodes and edges must be evenly distributed, edges should all have the same length, edges must be straight lines, isomorphic substructures should be displayed in the same manner, edge-crossings should be kept to a minimum, etc.<sup>3</sup> Trees have received the most attention in the literature. Consequently, additional aesthetics rules have also been formulated for them. For example, nodes with equal depth should be placed on a same horizontal line, distance between sibling nodes is usually fixed, etc. See again the book of Battista et al. [5] for further examples.

The Reingold and Tilford algorithm for trees [103], [121] (see Fig. 1) is a good example of a layout algorithm achieving these aesthetic goals. Isomorphic subtrees are laid out in exactly the same way and distance between nodes is a parameter of the algorithm. On the other hand, the more straightforward and naive algorithm for displaying a tree, consisting of distributing the available horizontal space to subtrees according to their number of leaves, actually fails to achieve some of the aesthetic rules listed above.

Although the adjective “aesthetic” is used, some rules were originally motivated by more practical issues. For instance, minimization of the full graph area might be an important criterion in applications. Some of the rules clearly apply to a certain category of graphs or layouts only, others have a more “absolute” character. Furthermore, each of the rules defines an associated optimization problem, used in a number of nondeterministic layout algorithms.

There has been some work lately which questions the absolute character of those rules, however. Usability studies were conducted in order to evaluate the relevance of these aesthetics for the end-user. Purchase [100] demonstrates that “reducing the crossings is by far the most important aesthetic, while minimizing the number of bends and maximizing symmetry have a lesser effect.” Her work concludes by prioritizing these aesthetics; see also Purchase et al. [101], [102] for more details. Other authors [10], [29], [86] report differences in the perception of a graph depending on its layout. Unfortunately, usability studies necessitate a great effort, both to realize the experimentation itself and to analyze its results properly, but we regard this line of work as essential for information visualization. Usability studies have recently gained credibility in the graph visualization community as well, recognizing their contribution to help focus on important issues in the area.

3. Actually, some aesthetics are quite arbitrary and are not seen as absolute rules any more [100], [101]. Ware’s book [123] is also an interesting source of information for this topic.

A wide variety of tasks related to graph drawing have been studied: layering a graph, turning it into an acyclic directed graph, planarization of a graph, minimizing the area occupied by a layout, minimizing the number of bends in edges, etc. Unfortunately, many of the associated algorithms are too complex to be practical for applications. On the positive side, this has motivated the development of effective heuristics to overcome the complexity of some of these problems [5], [34].

In graph visualization, a major problem that needs to be addressed is the *size* of the graph. Few systems can claim to deal effectively with thousands of nodes, although graphs with this order of magnitude appear in a wide variety of applications. NicheWorks [126], GVF [64], and H3Viewer [94] are among the few systems that claim to handle data sets with thousands of elements. The size of a graph can make a normally good layout algorithm completely unusable. In fact, a layout algorithm may produce good layouts for graphs of several hundred nodes, but this does not guarantee that it will scale up to several thousand nodes. For example, Fig. 1 illustrates a tree with a few hundred nodes laid out using the classical Reingold and Tilford algorithm. The high density of the layout comes as no surprise and changing particular parameters of the algorithm will not improve the picture for the graph. Other 2D layout techniques could be used, but most layout algorithms suffer from the same problem. Because the layout is so dense, interaction with the graph becomes difficult. Occlusions in the picture make it impossible to navigate and query about particular nodes. The use of 3D or of non-Euclidean geometry have also been proposed to alleviate these problems. Sections 2.4 and 2.5 provide more details about these techniques. However, beyond a certain limit, no algorithm will guarantee a proper layout of large graphs. There is simply not enough space on the screen. In fact, from a cognitive perspective, it does not even make sense to display a very large amount of data. Consequently, a first step in the visualization process is often to reduce the size of the graph to display. Classical layout algorithms remain usable tools for visualization, but only when combined with these techniques.

Other properties of a layout algorithm can be critical when navigating through a graph. The concept of *predictability* has been identified as an important and necessary aspect of layout algorithms [61], [99]. What is meant by predictability is that two different runs of the algorithm, involving the same or similar graphs should not lead to radically different visual representations. This property is also referred to in the literature as “preserving the mental map” of the user [90]. Predictability is often ignored during analysis of classical layout algorithms, which are usually used to produce a static view of a graph.

Another important issue is *time complexity*. Any visualization system needs to provide near real-time interaction, where updates must be done in very short time intervals in order to escape the notice of the user. Having an accurate estimate of the time complexity of an algorithm can be of great help for the implementation of large systems when planning which algorithm to apply.

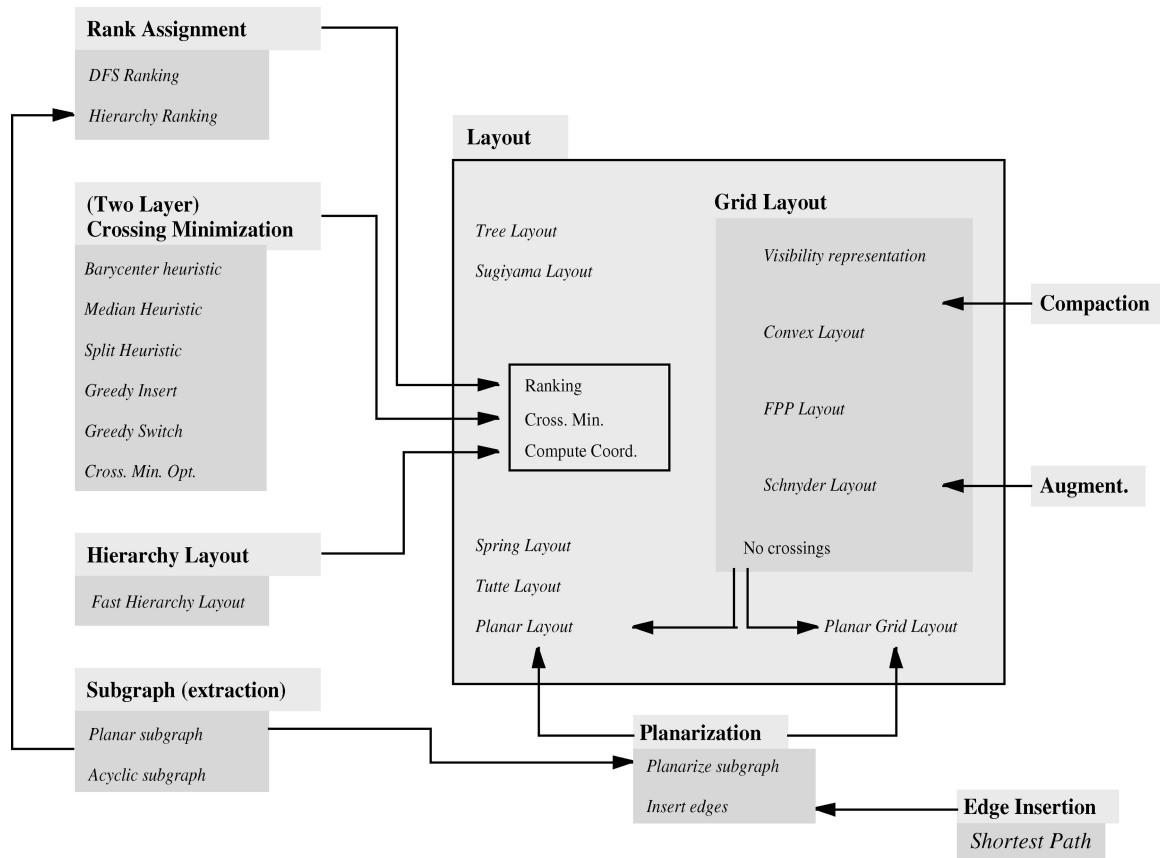


Fig. 2. Overview of graph layout algorithms. (Reproduced from Mutzel et al. [96], courtesy of T. Mutzel, Max-Planck-Institut, Saarbrücken, Germany.)

## 2.2 Traditional Layout—An Overview

We will briefly review existing layout techniques in graph drawing, keeping the issues of predictability and time complexity in mind. Fig. 2 gives a classification of existing layout techniques. This classification is the work of Mutzel et al. [96]. Most of the algorithms are described in the book by Battista et al. [5]. We will focus on the *Layout* box containing a list of possible layout types.

A classical *Tree Layout* will position children nodes “below” their common ancestor. The algorithm by Reingold and Tilford [103], [121] is probably the best known layout technique in the tree layout category (see Fig. 1). It can be adapted to produce top-down, as well as left-to-right tree layout, and can also be set to output grid-like positioning.

H-tree layouts are also classical representations for binary trees [113] which only perform well on balanced trees. Eades [35] suggests a variation of the algorithm that behaves well in general (see Fig. 3). The radial positioning by Eades [35] places nodes on concentric circles according to their depth in the tree (see Fig. 4). A subtree is then laid out over a sector of the circle and the algorithm ensures that two adjacent sectors do not overlap (although this condition can be ignored to obtain relatively good drawings on average [63], [126]). The cone tree [20], [106] algorithm can be used to obtain a “balloon view” of the tree by projecting it onto the plane [20], [71], where sibling subtrees are included in circles attached to the father node. It is also possible to compute the node position directly, without

using cone trees[87] (see Fig. 5; Section 2.4 describes cone trees in more detail).

The Reingold and Tilford algorithm produces a more classical drawing in the sense that the drawing clearly reflects the intrinsic hierarchy of the data. The radial and H-tree positioning are different in this respect because it is less clear where the root of the tree is and, thus, one might explore the graph in a less hierarchical fashion. The Reingold and Tilford, H-tree, radial, and balloon layouts are all predictable. Tree layout problems usually have the lowest complexity, which is linear in the number of nodes. As we can see, although the *Tree Layout* box occupies only a small area of Fig. 2, it contains a variety of layouts. Chapter 3.1 of the book by Battista et al. [5] is a good starting point for a further overview of these tree layout techniques. Two tree layout algorithms, which are not part of the “traditional” arsenal, are also worth mentioning here: tree-maps [72] (see Fig. 6) and onion graphs [115], which represent trees by sequences of nested boxes. It is important to note that, in tree-maps, the size of the individual rectangles is significant. For example, if the tree represents a file system hierarchy, this size may be proportional to the size of the respective file. This is why tree-maps enjoy popularity in information visualization in spite of the fact that it is difficult to perceive the structure in this representation.<sup>4</sup> An attempt to overcome this problem has

4. The value of the tree-map is demonstrated in an interactive java applet at <http://smartmoney.com/marketmap/>.

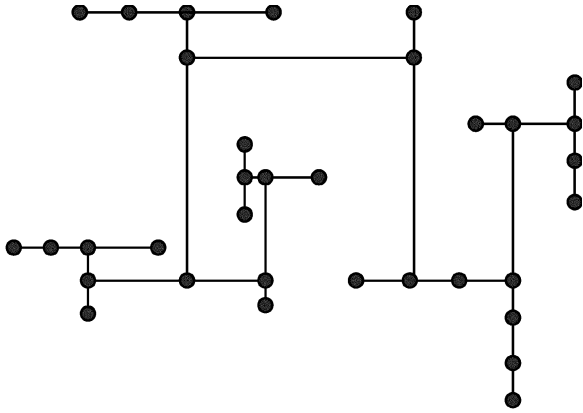


Fig. 3. H-tree layout.

been recently presented by Wijk and Wetering [125] in the form of cushion tree-maps.

A separate box at the bottom of Fig. 2 is devoted to *Planarity*. This is a critical issue in graph drawing because the planarity of a graph may be an important constraint imposed by practical applications (such as graphs representing printed circuit boards). The complexity for testing planarity for undirected graphs can be linear [67]. (See Chapter 3.3 in Battista et al. [5]. See also Mehlhorn and Mutzel [88] for a discussion on implementation issues.) However, many applications impose the additional requirement that edges are all in the same direction (planar drawings often make use of edges going around some nodes to avoid crossings). This condition, called *upward planarity*, turns the original problem into an NP problem. (See Garg and Tamassia [54]. See also Chapter 6 in Battista et al. [5]). In information visualization applications, it only makes sense to check for planarity when dealing with a small and sparse graph [3], [30], such as a subgraph obtained by clustering a larger graph (see Section 4.). In general, we can safely say that planarity is not a central issue in information visualization.

The *Sugiyama Layout* box included in Fig. 2 is named after the seminal work by Sugiyama et al. on layout for

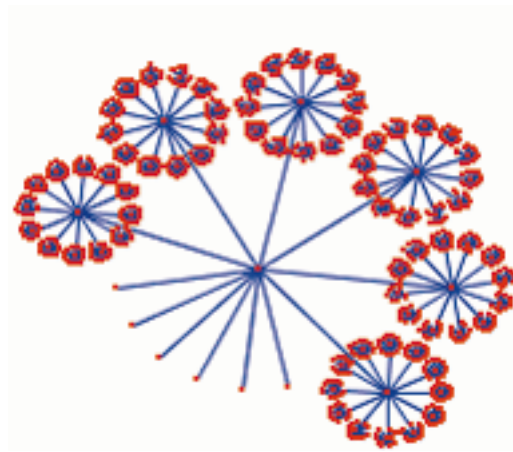


Fig. 5. Balloon view.

general directed graphs [117]. The basic approach to laying out a directed graph is to first decide on a *layering* of its nodes; that is, assign a layer number to each node and place nodes of a given layer in a certain order. Several layering techniques exist, the majority of which rely on the extraction of an acyclic subgraph. In this process, a subgraph containing all nodes of the original graph is extracted in such a way that, when nodes have been placed in their respective layers, edges will all point in the same direction (usually downward). Another solution is not to extract a subgraph, but to turn the original graph into an acyclic one by reversing the direction of a subset of the edges.

Once the nodes have been assigned to layers, one must position the nodes within the same layer following an imposed order. A major effort has been invested in edge-crossing minimization [5], [34] since the crossing of edges has been recognized as an obstacle to the readability of graphs [100], [101]. This is usually done by minimizing the number of edge-crossings between two consecutive layers. This minimization step is at the core of complexity for the whole algorithm. Note that these strategies do not address the problem of minimizing the number of crossings in the

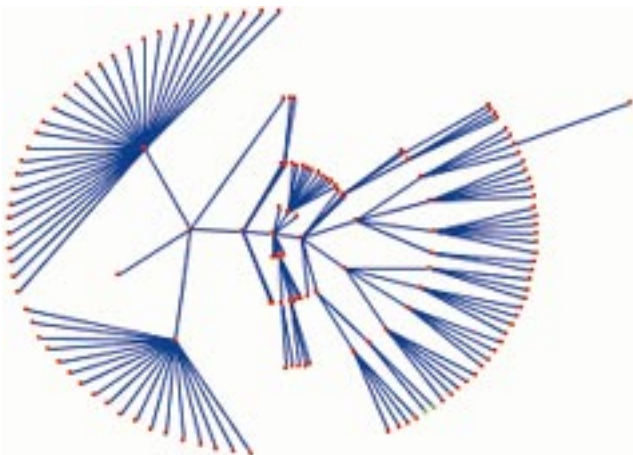


Fig. 4. Radial view.

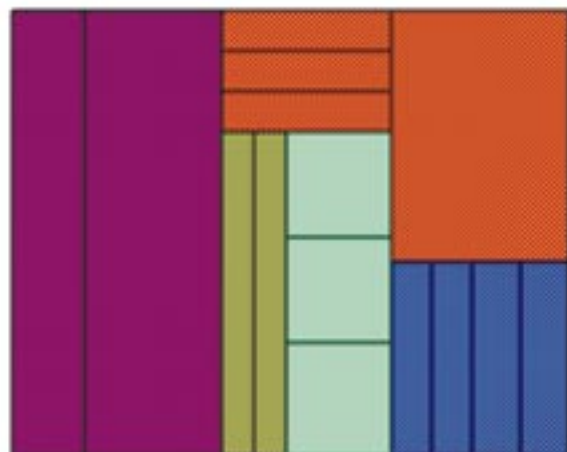


Fig. 6. Tree-map: rectangles with color belong to the same level of the (tree) hierarchy. (Adapted from Johnson and Schneiderman [72]).

whole graph: Even with the restriction of looking at consecutive layers only, minimization of edge-crossings is difficult and complex. In fact, Garey and Johnson proved the problem to be NP-hard [53] and Eades and Whitesides proved the corresponding decision problem to be NP-complete [36].

The complexity of a proper minimization has motivated the development of various heuristics for computing a good order for the nodes on a layer. Tutte [119] was the first to propose a heuristic: Starting from an order on the top and bottom layers, the coordinates of a node are defined to be the barycenter of those of its neighbors. This corresponds to the intuitive idea that a node should be kept “close” to its neighbors. The solution is then obtained by solving a system of linear equations. One variation to this scheme is to compute barycentric coordinates by performing a layer-by-layer descent in the graph. More generally, the four boxes on the left of the figure correspond to various preprocessing possibilities for the algorithm in the *Sugiyama Layout* category. New improvements and perspectives on the problem were published recently [73], [79], which include a detailed report on existing techniques [80] and a comparison of existing heuristics [81].

The critical element of the general scheme for directed graphs is its high complexity, although it might be kept within reasonable bounds if the size of the graph—or, should we say, subgraph—to be drawn is kept small. The ranking process in itself has a low cost. Indeed, a breadth first search of the graph returns an acyclic subgraph that can be used for layering. However, the choice of this subgraph can determine the quality of the final layout. We will return to that issue later. It is also not clear whether any algorithm in this class will be predictable. Some approaches can certainly be made predictable, but then the price to pay will be a greater complexity due to the loss in flexibility in reordering the nodes on a layer. Battista et al. give a detailed account of edge-crossing minimization in Chapter 9 of their book [5].

The *Spring Layout* box stands for all nondeterministic layout techniques, also called *Force-Directed Methods*. Eades [33] was the first to propose this approach in graph drawing, modeling nodes and edges of a graph as physical bodies tied with springs. Using Hooke’s law describing forces between the bodies, he was able to produce layouts for (undirected) graphs. Since then, his method was revisited and improved [28], [47], [49], [75]. Mathematically, the methods are based on an optimization problem. Different physical models lead to algorithms of different complexities and they produce layouts of varying quality. Spring layouts have been used successfully to produce well-balanced layout for graphs. In some cases, their output can even behave well with respect to edge-crossing minimization without any supplementary efforts [47]. Bertault has recently developed a force-directed model preserving edge-crossings, turning it into a more predictable approach [9].

In general, however, force-directed methods can be rather slow. Each iteration involves a visit of all pairs of nodes in the graph and the quality of the layout depends on the number of full iterations: each step improves the positions following the underlying mathematical model.

Even one of the best variants [47] is still estimated to work with a complexity of  $O(N^3)$ , where  $N$  is the number of nodes in the graph. Moreover, two different runs of the algorithm on almost identical graphs might produce radically different layouts. In other words, the methods may be highly unpredictable. This makes them less interesting for information visualization since unpredictability can be a major problem for interaction. However, in some cases, the lack of predictability can be compensated for if the graph is small or sparse, by animating changes in the layout to help the user in adapting to the new drawing [69]. For further information on force-directed methods, the reader should refer to the comparison of nondeterministic techniques of Brandenburg et al. [12] or Chapter 10 in the book of Battista et al. [5].

We will not discuss layouts on grids here. We refer to Battista et al. [5] for details on that, as well as for learning more about the additional techniques included in the boxes “Compaction” and “Augmentation” on the right side of Fig. 2. None of these techniques play a central role in graph visualization.

The classification of algorithms in Fig. 2 assumes that layout is determined only by the nodes and edges, without additional constraints. However, some work has been done with applications where the nodes of the graph have preassigned positions in the plane, such as geographical positions. The challenge is then to find a way to draw edges, for example, by using polylines or spline curves [6], [13], [97].

## 2.3 Spanning Trees

A general problem with the majority of the available techniques is that they are only applicable for relatively small graphs.<sup>5</sup> The “traditional” concerns of Graph Drawing become much less relevant in graph visualization, which typically deals with relatively large graphs. In general, it makes no sense to test a graph of several hundred nodes for planarity or to try to minimize edge-crossings. Often, the most obvious and practical solution is simply to layout a spanning tree for the graph. As we have already seen, tree layout algorithms [20], [35], [103], [121] have the lowest complexity and are simpler to implement. The problem is then transformed into one of finding a spanning tree. That option involves laying out a graph based on the positioning of a tree containing all nodes of the graph which had been previously extracted from the graph. Additional edges are then added to the tree. The literature in graph theory proposes a long list of algorithms to compute spanning trees for graphs, both for the directed and undirected cases (see, for example, Jungnickel [74]). Incidentally, using a spanning tree to layout a graph can also be a solution to gain predictability of the layout. Although spanning trees are obviously not the only layout approach in graph visualization, they certainly do and will play an important role.

Extracting a spanning tree with no particular property can be done easily. One approach is to visit the nodes of the

5. This is clearly shown by the size of the graphs submitted each year to the Graph Drawing Contest, although bigger graphs—and, also, graphs coming from real-world life situations applications—have also been included in recent years.

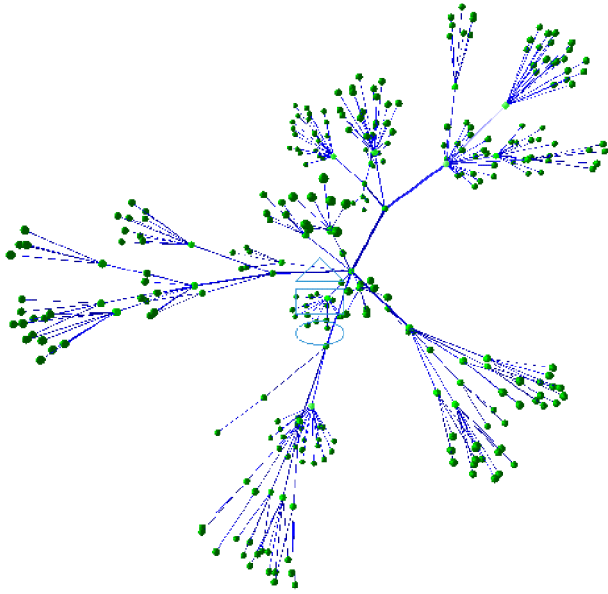


Fig. 7. 3D version of a radial algorithm. (Courtesy of S. Benford, University of Nottingham, U.K.)

graph through a breadth first search and collect edges to form a tree. The search can start from a node that is more likely to “act” as the root of the extracted tree. A node whose distance to all other nodes is minimal is a good candidate [11]. More sophisticated algorithms have been designed to satisfy various optimization goals. If a weight function exists for the graph, algorithms exist to compute spanning trees minimizing (or maximizing) the total weight of the tree. One solution is to iteratively build a tree by adding edges adjacent to the set of already selected nodes, each time selecting an edge with minimal (maximal) weight. Different choices for the weight function will yield different solutions and will also affect the complexity of the extracting process (see, for example, Chapters 4 and 5 of Jungnickel [74]). The complexity of this task varies according to the variant used. The naive solution has a complexity of  $O(N^2)$ ; better solutions exist which bring the complexity down to  $O(N \log N)$  or to  $O(E \log N)$  (where  $N$  and  $E$  denote the number of nodes and edges of the graph, respectively).

A weight function can be used to extract different spanning trees and, consequently, to obtain different possible layouts for the same graph (although the implementor must be aware of the fact that a spanning tree realizing an optimization goal for a given weight function does not necessarily produce a good view of the graph). Use of weight functions can also be applied to directed acyclic graphs to avoid going through the task of edge-crossing minimization. For large and dense acyclic directed graphs, the use of layers as a weight function (the weight of a node or edge is its layer number) has proven to give good results (see, for instance, Herman et al. [63]).

## 2.4 3D Layout

One popular technique is to display graphs in 3D instead of 2D. The hope is that the extra dimension will give, literally, more “space” and that this will ease the problem of

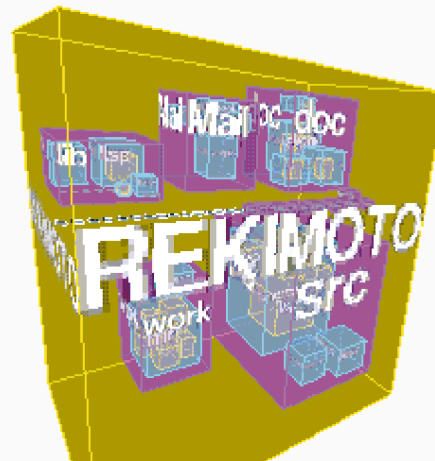


Fig. 8. Information Cube. (Courtesy of J. Rekimoto, Sony Computer Science Laboratory, Inc., Japan [104].)

displaying large structures. Furthermore, the user can navigate to find a view without occlusions. The simplest approach is to generalize classical 2D layout algorithms for 3D. Fig. 7, for example, shows a 3D version of a radial tree algorithm, while Fig. 8 is a generalization [104] of the two-dimensional approach using nested boxes [115]. Most force-directed methods are also described in dimension independent terms, which allows them to be generalized to 3D (such as the approaches based on simulated annealing by Davidson and Harel [28] and, also, from Cruz and Twarog [27]). The reader may find further examples in the overview by Young [128] or in the new book by Ware [123].

In spite of their apparent simplicity, Figs. 7 and 8 show that displaying graphs in 3D can also introduce new problems. Objects in 3D can occlude one another and it is also difficult to choose the best “view” in space [38]. As a consequence, virtually all 3D displays of graphs include additional visual cues, like transparency, depth queuing, etc. They also allow the user to interactively change the view by “moving around” in space. But, the ability to change perspective adds another difficulty. Common practices, such as the minimization of edge-crossings, are less rewarding if the user can change the perspective and see edge-crossings from another angle. However, it is the job of the application to provide the best possible view of the information in the perspective initially provided to the user, so aesthetics cannot be dismissed.

The cone tree, [107] (see Fig. 9) is one of the best known 3D graph (in this case, tree) layout techniques in information visualization.<sup>6</sup> In contrast to the previous examples, cone trees have been developed directly for 3D, instead of generalizing another 2D algorithm.

Mathematically, the layout is quite simple. A node is placed at the apex of a cone with its children placed evenly along its base. In the original implementation, each layer has cones of the same height and the cone base diameters for each level are reduced in a progression so that the bottom layer fits into the width of what the authors called

6. The term “cam tree” is also used. Strictly speaking, cam trees are horizontal arrangements, whereas cone trees are vertical. We will not differentiate between them.

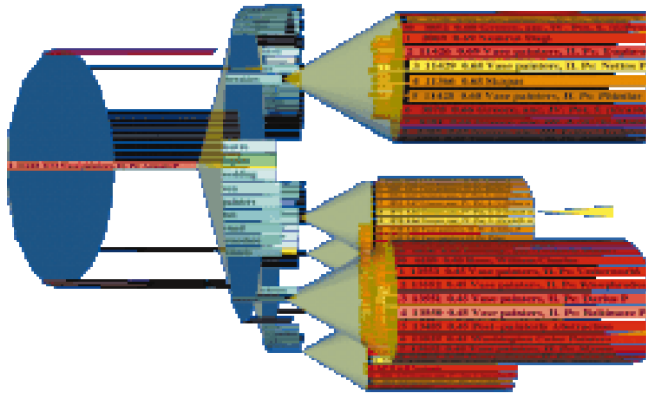


Fig. 9. A cone tree. (Courtesy of M. Hemmje, GMD, Germany [59].)

the “room,” i.e., the box containing the full cone tree. The original idea of cone trees has been reimplemented by others [20], [59], [71] with, in some cases, a refined layout algorithm. Carrière and Kazman [20], for example, calculate an approximation of the diameter for each cone base by traversing the tree bottom-up and by taking the number of descendents into account at each step to make better use of the available space. Jeong and Pang [71] replace the cones with discs to reduce occlusion.

The interactive and visual aspects of cone trees are essential to make them usable. Not only are some of the labels at the nodes transparent, but the user can pick any node and rotate the cone tree so that the chosen node is brought to the front. This can either be done automatically, by the system, or as a result of further user interaction. For horizontal cone trees, the effect somewhat resembles stepping through Rolodex cards arranged in multiple levels.

Gaining more “space” is not the only possible advantage of using 3D. Because of general human familiarity with 3D in the physical world, 3D lends itself to the creation of real-world metaphors that should help in perceiving complex structures. One of the earliest widespread applications is the File System Navigator (see Fig. 10), which came with earlier SGI Workstations until version 5 of their operating system. The layout of the graph (a tree representing the user’s file space) is a simple planar layout. The 3D aspect consists, on the one hand, of adding blocks on the plane whose sizes are proportional to the file sizes and, on the other hand, of the ability to “fly” over the virtual landscape created by those blocks. This cityscape approach has been implemented in various other systems, see, for example, SDM [24], or, more recently, the system presented by Chen and Carr [22]. More complex 3D metaphors include the Perspective Wall [107], which represents the data as posters on a big wall in virtual space. VizNet [43] and Vitesse [98] both use an idea similar to the perspective wall by mapping objects onto the surface of a sphere with highly related objects placed close to a selected object of interest. The Web Book [15] displays an animated book in 3D with Web page

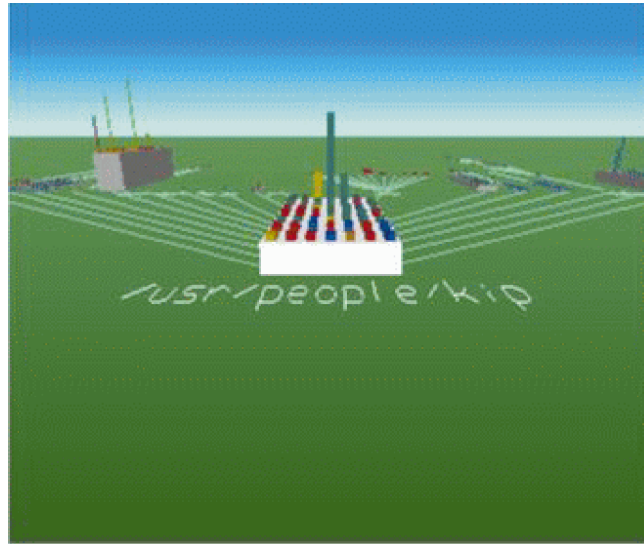


Fig. 10. The SGI File System Navigator.

contents, etc. Here again, we refer to the overview of Young [128] for further examples.

In spite of all the technical development in the area, and their undeniably attractive features, 3D graph visualization techniques have significant difficulties. In our view, the main reason lies with the inherent cognitive difficulties of 3D navigation in our current systems. Perceptual and navigational conflicts are caused by the discrepancy of using 2D screens and 2D input devices to interact with a 3D world, combined with missing motion and stereo cues (see the overview of Ware and Franck [122] for how important these cues are). Limited 3D interaction, such as the ability to rotate an object for inspection without getting closer to it, may provide 3D interaction that doesn’t cause disorientation. If advanced VR-like systems, such as a Workbench, CAVE, or large tiled displays are used, some of these difficulties may be solved. However, such facilities are not widely available and are still too expensive to serve as a basis for most information visualization applications. When more advanced display and interactive facilities (e.g., haptic displays and interaction, stereo views, etc.) become more widely available, 3D techniques may have a profound effect in graph visualization.

## 2.5 Hyperbolic Layout

The hyperbolic layout of graphs (mainly trees) is one of the new forms of graph layout which has been developed with graph visualization and interaction in mind. The first papers in this area are from Lamping et al. [82], [83], followed by a series of papers by Munzner [92], [93], [94]. Both developed, for example, Web content viewers based on these techniques. The technique has been since used by other systems, too, see, for example, Robinson [108] or Wilson and Bergeron [127]. Hyperbolic views, which can be implemented in either 2D or 3D, provide a distorted view of a tree (see Fig. 11). It resembles the effect of using fish-eye lenses on traditional tree layouts. This distorted view makes it possible to interact with potentially large trees, making it suitable for real-life applications. We will come back to this

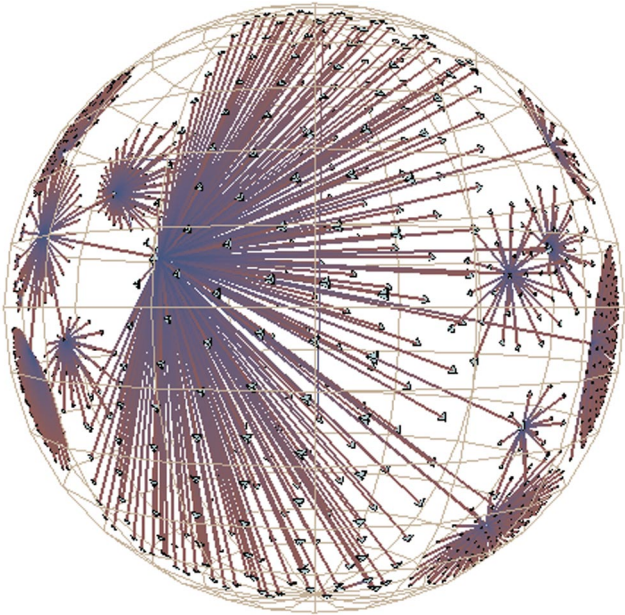


Fig. 11. Hyperbolic view of a tree in 3D. (Courtesy of T. Munzner, Stanford University.)

distortion effect later in this survey (see Section 3.2) when we will focus on navigation rather than layout.

Hyperbolic views represent a radically different direction in layout when compared to the algorithms described so far, due to their different geometrical background. In fact, some of the classical layout algorithms can be reused in a hyperbolic setting, yielding sometimes quite different results, as demonstrated later in this section. Hyperbolic views are also surrounded by a sort of mystery because few people in the information visualization community really understand the mathematics of hyperbolic visualization. Furthermore, it is quite difficult to reproduce the results. Unfortunately, none of the papers are didactic enough to reveal the mystery. We will discuss the main elements of these layout methods further with the hope that the reader will gain a better understanding of the technique.

Hyperbolic geometry is based on an axiomatic system almost identical to the traditional Euclidean axioms with the exception of one, the so-called fifth postulate. Whereas the Euclidean postulate states that if a line does not intersect a point, then there is *only one* line intersecting the point and parallel to the original line (i.e., nonintersecting and coplanar), in hyperbolic geometry there exists *more than one* such parallel line. This alternative set of axioms results in a perfectly consistent form of geometry, albeit different in flavor: The traditional trigonometric equations are no longer valid, the sum of the internal angles of a triangle is no longer 180 degrees, etc.<sup>7</sup> (These differences, by the way, represent significant difficulties for implementors using hyperbolic geometry.)

It is also possible to define a consistent *model* for the hyperbolic plane (or space) within the Euclidean space, thereby making a logical link between the two worlds. A model in this respect means defining a subset of the

7. The interested reader might want to refer to Coxeter [25] for further details. Also, look at the papers of Gunn [56] or Hausman et al. [57].

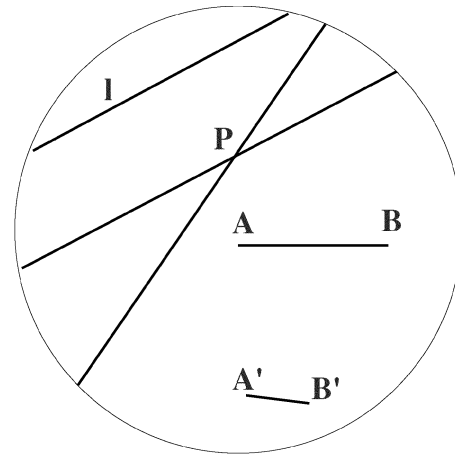


Fig. 12. The Klein model for the hyperbolic plane. The line segment  $AB$  and  $A'B'$  have an equal length in the hyperbolic sense.

Euclidean space and the notions of “points,” “lines,” “intersections,” “length” within this subset so that the axioms of hyperbolic geometry would be valid locally. Several different models were developed. The best known are the Klein and the Poincaré models. The Klein model (see Fig. 12) uses an open disc (or sphere for 3D) as a subset, i.e., the hyperbolic plane in this model consists of the points within the perimeters of the disc. Hyperbolic lines are represented by chords of the disc. Intersection is just the Euclidean intersection. The only major difference is the *length* of a line segment. We will not give a detailed definition here. Suffice it to say that this length is defined as a function of the position of the points vis-à-vis the perimeter of the disc: Segments which are congruent in a hyperbolic sense are exponentially smaller in the Euclidean sense when approaching the perimeter. To prove the local validity of *all* the axioms of hyperbolic geometry requires some nontrivial work. The validity of the negation of Euclid’s fifth postulate is quite obvious, though, just consider the line  $l$  and the point  $P$  on the figure. The Poincaré model is quite similar, although hyperbolic lines are represented by arcs which intersect orthogonally the perimeter of the disc.

It is now possible to give a more exact description of what the hyperbolic graph layouts do orthogonally: They perform a layout algorithm in the hyperbolic plane or space and, then, display the results in the familiar Euclidean plane or space *using one of the models of hyperbolic geometry*. That is, what we see is *not* hyperbolic geometry per se, but its representation in Euclidean geometry. The original paper of Lamping et al. used the Poincaré model, whereas Munzner primarily uses the Klein model. In Fig. 11, for example, the Klein model for hyperbolic 3D space is used to display the tree. The distortion effect referred to earlier is the result of the exponential shrinking of congruent line segments closer to the disc perimeter when viewed in the Euclidean space.

The different spatial nature of hyperbolic geometry makes some rather simple layout algorithms suddenly viable. As an example, consider the outline of the following tree placement algorithm (see Fig. 13).<sup>8</sup> The algorithm starts

8. This algorithm is essentially the same as the one used in the paper of Lamping and Rao [83].



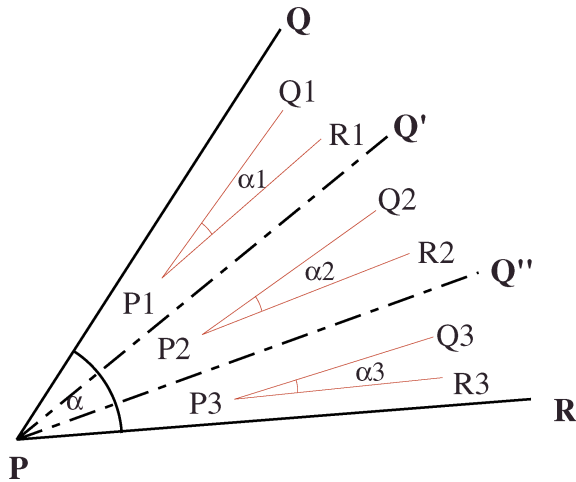


Fig. 13. A simple tree positioning algorithm on the Euclidean plane.

from the root of the tree, positioning the subtrees recursively in a circular fashion. In each step, the algorithm determines a wedge to place a subtree. The goal is to find wedges in such a way that no crossing would occur between edges of different subtrees. If the point  $P$  on the figure refers to a node, and the wedge  $QPR$  with angle  $\alpha$  is the one assigned to the subtree starting at  $P$ , the main step of the algorithm is to define subwedges for the subtrees of  $P$  (starting at  $P_1$ ,  $P_2$ , and  $P_3$ ). The angle  $\alpha$  is divided into (for the sake of simplicity, equal) subangles, one for each subtree. The subdivision of the original wedge results in the radii  $PQ'$ ,  $PQ''$ , etc. (see the figure). The points  $P_1$ ,  $P_2$ ,  $P_3$  are positioned in the middle of these subwedges at some suitable distance from  $P$ . The next step is to determine the constraining wedges for these subtrees. This can be done by establishing parallel lines with  $PQ$ ,  $PQ'$ ,  $PQ''$ , starting at the points  $P_1$ ,  $P_2$ ,  $P_3$ , etc. These lines will determine the new wedges with angles  $\alpha_1$ ,  $\alpha_2$ ,  $\alpha_3$ , etc., and the recursion step can continue for each of the corresponding subtrees. Obviously, because parallel lines are used, the children's wedges will not overlap.

The algorithm is very naive and would lead to quite unusable figures on the Euclidean plane. Indeed, the wedge angles become very small after a few steps, which shrinks the space available for the next subtree. However, if the same algorithm is used on a hyperbolic plane, the situation is quite different. Fig. 14 shows the same algorithm in the Klein model. The major difference is the way the parallel lines to  $PQ'$ ,  $PQ''$ , etc., are calculated: The (hyperbolic) parallel lines are the lines intersecting *on* the perimeter of the disc of our model. The effect will be to "open" the angles  $\alpha_1$ ,  $\alpha_2$ ,  $\alpha_3$ . To cite Lamping and Rao [83]: "Each child will typically get a wedge that spans about as big an angle as does its parent's wedge." Of course, although visible on the Klein model, this statement has to be substantiated through explicit formulae using the hyperbolic trigonometric calculations, which is quite possible. The result is a perfectly feasible layout algorithm. It should be noted that Munzner uses different layouts. More details on her spherical placement can be found in one of her papers [93], which is actually a generalization of the cone tree

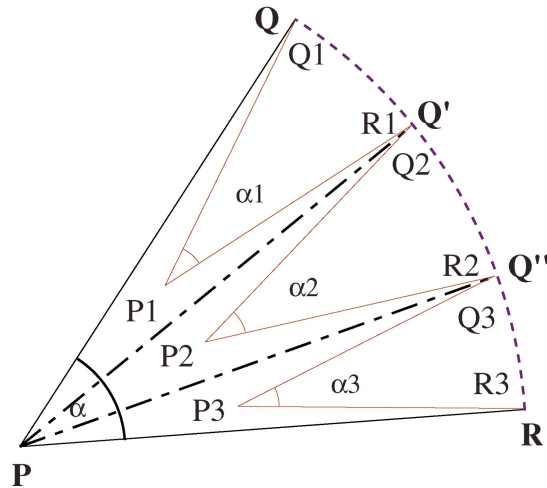


Fig. 14. The same tree positioning algorithm on the hyperbolic plane, using the Klein model to visualize the results.

algorithm described in Section 2.4. However, here again, the placement algorithm is used in terms of hyperbolic geometry, taking advantage of the "large space" available in hyperbolic space.

### 3 NAVIGATION AND INTERACTION

Navigation and interaction facilities are essential in information visualization. No layout algorithm alone can overcome the problems raised by the large sizes of the graphs occurring in the visualization applications. Furthermore, the task of revealing the *structure* of the graph calls for innovative approaches.

#### 3.1 Zoom and Pan

Zoom and pan are traditional tools in visualization. They are quite indispensable when large graph structures are explored. Zoom is particularly well-suited for graphs because the graphics used to display them are usually fairly simple (lines and simple geometric forms). This means that zoom can, in most cases, be performed by simply adjusting screen transformations and redraw the screen's contents from an internal representation, rather than zooming into the pixel image. In other words, no aliasing problems occur.

Zooming can take on two forms. *Geometric zooming* simply provides a blow up of the graph content. *Semantic-zooming* means that the information content changes and more details are shown when approaching a particular area of the graph. The technical difficulty in this case is not with the zooming operation itself, but rather with assigning an appropriate level of detail, i.e., a sort of clustering, to subgraphs. The more general problem of clustering is addressed in Section 4.

Although conceptually simple, zoom and pan does create problems when used in interactive environments. Let us imagine, for example, the following setting: The graph being displayed is the road network of Europe and the user has zoomed into the area around Amsterdam. The user then wants to change the view of the area around Milano. Doing this without changing the zoom factor, at

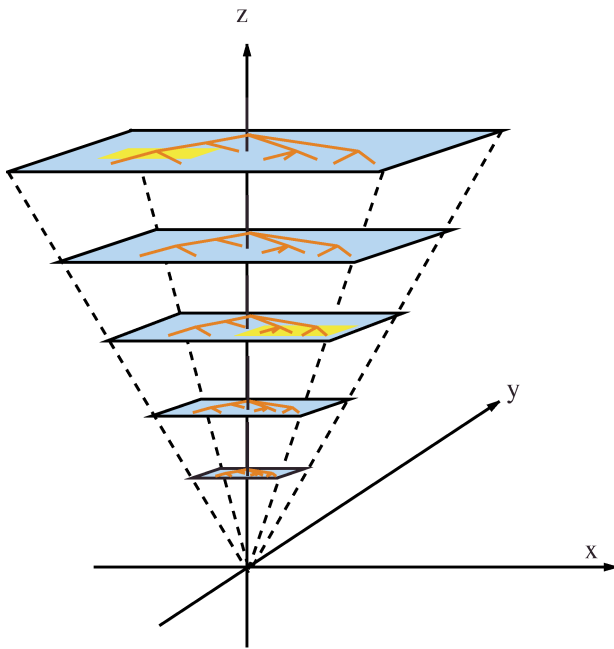


Fig. 15. A space-scale diagram. The yellow rectangles represent possible window positions in space-scale, yielding different zoom factors and pan positions. (Adapted from Furnas and Bederson [51].)

least temporarily, might be too slow because the user has to first zoom out, pan to Milano, and zoom in again. Furthermore, the user wants the system to make the necessary moves smoothly. A naive implementation might calculate the necessary changes for the pan and the zoom independently and perform the changes in parallel. The problem is that, when zooming in, the world view expands exponentially fast and the target point moves away faster than the pan can keep up with. The net result is that the target is approached nonmonotonically: It first moves away as the zoom dominates and only later comes back to the center of the view, which can be quite disturbing.

The zoom and pan problem is not restricted to graphs nor is the elegant solution proposed by Furnas and Bederson [51] to alleviate it. Nevertheless, graph visualization systems can greatly benefit from their approach, so we will provide a short description here. Furnas and Bederson introduce the concept of space-scale diagrams (see Fig. 15). The basic idea is to define an abstract space “by creating many copies of the original 2D picture, one at each possible magnification, and stacking them up to form an inverted pyramid.” Points in the original image can be represented by rays that contain information both about the point and its magnification. Various combinations of (continuous) zoom and pan actions can then be described as paths in this space by describing the central position of a window parallel to the  $x$ - $y$  plane. A cost, or “length,” can also be associated to each path and, if the length is judiciously chosen, a minimum length path can represent an optimal combination of zoom and pan movements. Furnas and Bederson not only give a solution to the problem outlined above; space-scale diagrams can also be used to describe semantic zooming (instead of stacking the same picture in the pyramid, the content of the picture may depend on

the magnification level), which also allows for the development of a specialized authoring system for semantic zooming [52].

## 3.2 Focus+Context Techniques

A well-known problem with zooming is that if one zooms on a focus, all contextual information is lost.<sup>9</sup> Such a loss of context can become a considerable usability obstacle. A set of techniques that allow the user to focus on some detail *without* losing the context can alleviate this problem. The term *focus+context* has been used to describe these techniques. They do not replace zoom and pan, but rather complement them. The complexity of the underlying data might make zoom an absolute necessity. However, focus+context techniques are a good alternative and full-blown applications systems often implement both.<sup>10</sup>

### 3.2.1 Fisheye Distortion

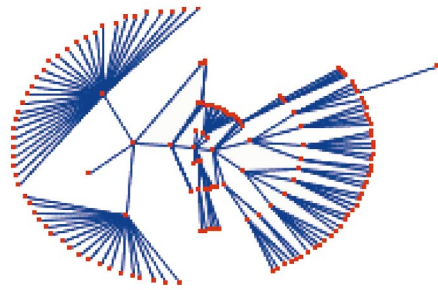
Graphical fisheye views are popular techniques for focus+context. Fisheye views imitate the well-known fisheye lens effect by enlarging an area of interest and showing other portions of the image with successively less detail (see Fig. 16).

We will describe some of the mathematics involved in the fisheye technique. Conceptually, the graph is mapped onto the plane and a “focus” point is defined (usually by the user). The distance from the focus to each node of the tree is then distorted by a function  $h(x)$  and the distorted points, and connecting edges, are displayed. The function  $h(x)$  should be concave, mapping monotonically the  $[0, 1]$  interval onto  $[0, 1]$  (see Fig. 17). The distortion created by the fisheye view is the consequence of the form of the function, which has a faster increment around 0 (hence affecting the nodes around the focus), with the increment slowing down when closing up to 1. The exact definition of the function may yield a lesser or stronger distorting effect. A simple distortion function, for example, used by Sarkar and Brown [110], [111] is:  $h(x) = (d + 1)/(d + 1/x)$  (this is the function plotted in Fig. 17). The factor  $d$  is the so-called distortion factor, which can be set interactively by the user. It should be positive; the larger it is, the stronger the fisheye distortion. Fig. 18 shows the effect of this function (with  $d = 4$ ) on the regular grid around the origin.

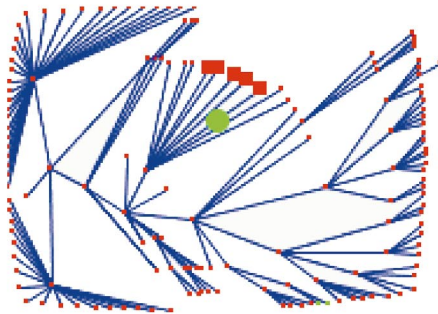
There are some variations to this basic scheme. What we have just described is usually referred to as a “polar” distortion, in the sense that it applies to the nodes radially in all directions starting from the focus point. An alternative is to use a “Cartesian” fisheye: The distance distortion is applied independently on the  $x$  and  $y$  directions before establishing the final position of the node (see again Fig. 16). Other variations are possible. Consult the overview of Carpendale et al. [18] or Keahey and Robertson [77] for further examples and for their visual effects. The final

9. Unless a separate window, for example, keeps the context visible, which is done by several systems. But, this solution is not fully satisfactory either.

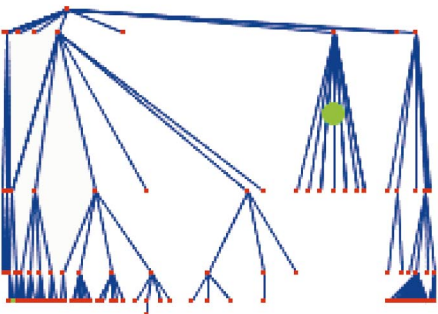
10. All techniques described in this section are *geometric*, i.e., they operate on the geometric representation of the underlying graphs. This is in contrast with a *logical* focus+context view described in an often-cited paper of Furnas [50]. In our view, the work of Furnas is more related to what we call “metrics,” rather than to graphical focus+context. See Section 4.2 for further details.



(a)



(b)



(c)

Fig. 16. Fisheye distortion. (a) Represents the graph without the fisheye. (b) Uses polar fisheye, whereas (c) uses Cartesian fisheye with a different layout of the same graph. The green dots on (b) and (c) denote the focal points of the fisheye distortion. Note the extra edge-crossing on (b).

choice should depend on the style of the graph to be explored, as well as the layout algorithm in use.

This simple but powerful technique is an important form of navigation that complements zoom and pan. However, implementors should be aware of one of the pitfalls. The essence of a fisheye view is to distort the position of each node. If the distortion is faithfully applied, the edges connecting the nodes will also be distorted. Mathematically, the result of this distortion is a general curve. Standard graphics systems (e.g., X11, Java2D, OpenGL) do not offer the necessary facilities to transform lines into these curves easily (the curves can be rather complex). The implementer's only choice is, therefore, to approximate the original line segments with a high number of points, transform

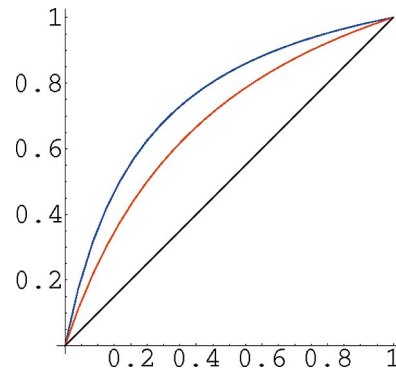


Fig. 17. The Sarkar-Brown distortion function with a distortion factor 2 (red curve) and 4 (blue curve).

those points, and display a polyline to approximate the ideal, transformed curve. The problem is that the number of approximating points must be relatively high if a smooth impression is desired (on average, 60 points per edge), which leads to a prohibitively large number of calculations and may make the responsiveness of the system sink to an unacceptably low level. The only viable solution is to apply the fisheye distortion on the node coordinates only and to connect the transformed nodes by straight-line edges. The consequence of this inexact solution is that unintended edge-crossings might occur (see, for example, the upper left quadrant of Fig. 16b). This is one of those typical situations when the pragmatism required by information visualization should prevail. If large graphs are explored, these extra intersection points do not really matter much and it is more important to keep the exploration tool fast.

### 3.2.2 Focus+Context Layout Techniques

The fisheye technique is independent of the layout algorithm and is defined as a separate processing step on the graphical layout of the graph. Interacting with fisheye means changing the position of the focus point and/or modifying the distortion value. This independence has positive and negative aspects. On the positive side, it allows for a modular organization of software in which fisheye is a separate step in the graph rendering pipeline somewhere between the layout module and the actual display. Fisheye can also be significantly faster than the layout algorithm, which is an important issue for interaction.

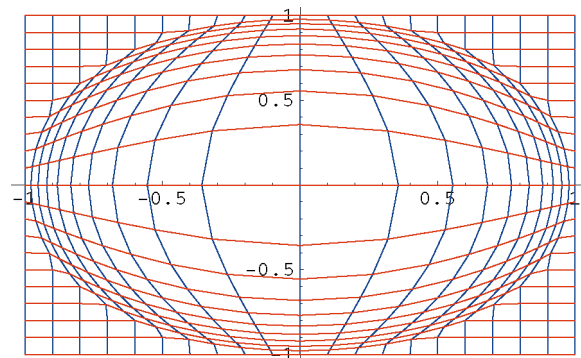


Fig. 18. Fisheye distortion of a regular grid of the plane. The distortion factor is 4.

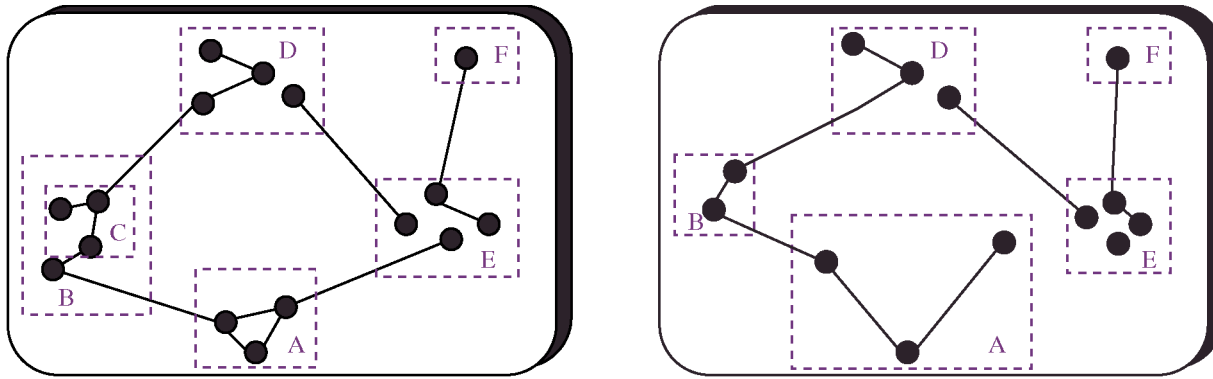


Fig. 19. Multifocal fisheye/zoom in a hierarchically clustered graph. The dotted rectangles denote the (logical) clusters. Note the disappearance of cluster C on the righthand side. (Adapted from Schaffer et al. [112].)

However, the fisheye distortion may destroy the aesthetics governing the layout algorithm. For example, as we have seen in the previous section, it can add new and unwanted edge-crossings.

An alternative is to build appropriate distortion possibilities into the layout algorithm itself, thereby merging the focus+context effects and the layout proper. Interacting with the distortion would mean interacting with (some) parameters governing the layout algorithm. The hyperbolic layout (see Section 2.5) does just that. The hyperbolic view of a graph, whether in 2D or 3D, produces a distorted view, not unlike the fisheye view (see Fig. 11). The equivalent of the focal point of the graphical fisheye view is the center of the Euclidean circle (or sphere) which is used to “map” the hyperbolic view onto the Euclidean space through either the Klein or the Poincaré model. Interacting with the view means changing the position of this center point within the graph.

Similar effects can be achieved by using 3D techniques (see also Section 2.4). By putting objects on 3D surfaces, for example, the view created by the perspective or parallel projections create a natural distortion on the 2D screen. In the Vitesse system [98], for example, the user has only limited 3D navigation facilities. The main goal of mapping objects onto a sphere or an ellipsoid is indeed to achieve a focus+context distortion. More complex surfaces (such as 3D surfaces of blended Gaussian curves) have also been used to achieve focus+context effects (see Carpendale et al. [17], [18]). Other 3D visualization techniques, already cited in Section 2.4 (such as the Perspective Wall [107]), apply this principle as well.

The hyperbolic layout is special because it is a graph layout algorithm that was developed with the focus+context distortion in mind. In fact, we do not know of any systematic research conducted on the existing, and more traditional, layout algorithms to decide whether such layout dependent distortions are possible or not, and, if yes, to exploit this feature in real systems. This is in spite of the fact that, at least in some cases, the possibility of applying such distortion control is clearly available. For example, Fig. 5 shows a balanced view of a tree, using a balloon layout algorithm [87]. This algorithm defines the radii of the circles by taking the number of descendents into account. The algorithm can be easily directed to give one of the circles a

larger “share” of the display space by shrinking all the others, thereby creating a focus+context effect on that circle [63]. We think that such research would provide valuable input for the implementors of graph visualization systems.

### 3.2.3 Further Issues in Focus+Context Techniques

There are further issues in the area of focus+context that can be of interest, some of which could be the basis for future research as well (a general characterization and taxonomy of distortion techniques is also presented in Leung and Apperly [84]). For example, fisheye is based on the choice of a distortion function, but we presented only a simple version here, used by Sarkar and Brown. This function can be replaced by others with different distortion features (arctan or tanh functions, piecewise linear approximations to speed up processing, etc.) [44], [77], [111]. The techniques can also be extended to 3D [19]. Also, just as we could speak about “semantic zoom,” one could also refer to “semantic focus+context,” meaning that, when the distortion becomes too “extreme,” in some sense, nodes might disappear after all. Sarkar and Brown describe this technique in their paper [110], but finer control over this facility might lead to new insights as well. Note that the space-scale diagrams [51] (see Section 3.1) can also be used to model fisheye distortions, which may lead to interesting results in combining (semantic) fisheye with zoom and pan. Finally, multifocal focus+context methods can also be applied [18], [76], [77], allowing the user to simultaneously concentrate on several important areas of the graph or to use the system in a cooperative environment [98].

An interesting example that combines various techniques, including multifocal zoom and focus+context, is provided by Schaffer et al. [112]. Their system also shows the fundamental importance of clustering, which we address in Section 4. They consider graphs that already have a hierarchical clustering. The lefthand side of Fig. 19 shows a drawing of the initial graph. The dotted rectangles denote the logical clusters (they appear on the figure only for the sake of the explanation; they would not necessarily appear on a real screen). The righthand side of the same figure shows the same graph after a multifocal zoom/fisheye action on clusters A and D. These clusters are now bigger, while the other clusters have shrunk. Moreover, cluster C has disappeared as a result of a sort of a “semantic fisheye” action on the graph. Schaffer et al. describe the

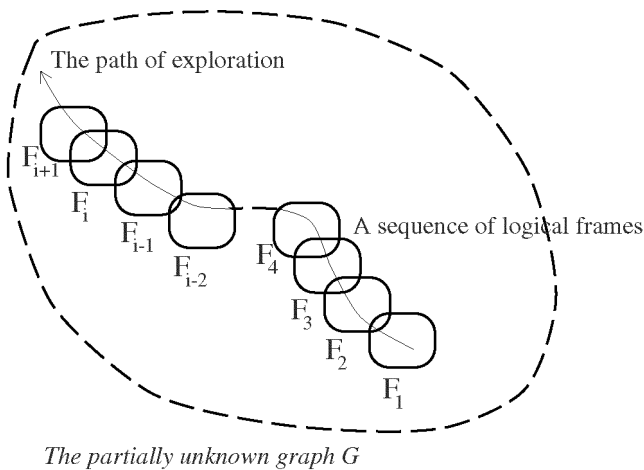


Fig. 20. Exploration of a huge graph. (Adapted from Huang et al. [69].)

mathematics of distortion and shrinking used to achieve these results. Similar ideas can also be found in the DA-TU system of Huang and Eades [70]. However, much remains to be done in combining these different approaches to achieve a coherent set of navigation techniques.

### 3.3 Incremental Exploration and Navigation

We have emphasized several times that the size of the graph is a major problem in graph visualization applications. There are cases when this size is so huge that it becomes impossible to handle the full graph at any time; the World Wide Web is an obvious example. *Incremental exploration techniques* are good candidates for such situations. The system displays only a small portion of the full graph and other parts of the graph are displayed as needed. The advantage of such an incremental approach is that, at any given time, the subgraph to be shown on the screen may be limited in size, hence, the layout and interaction times may not be critical any more. This approach to graph exploration is still relatively new, but interesting results in the area are already available, see, for example [14], [40], [68], [69], [99], [130].

Incremental exploration means that the system places a visible “window” on the graph, somewhat similar to what pan does. Exploration means to move this window (also referred to as *logical frames* by Huang et al. [68]) along some trajectory (see Fig. 20). Implementation of such incremental exploration has essentially two aspects, namely:

- decide on a strategy to generate new logical frames
- reposition the content of the logical frame after each change.

Generating new logical frames is always under the control of the user. In some cases, the logical frame simply contains the nodes visited so far. This is the case, for example, in the NESTOR tool, implemented by Zeiliger [130], which uses incremental exploration to record a history of the user’s surfing the World Wide Web: Newly accessed web pages are simply added to the logical frame to generate a new one. Huang et al. [68] (who also implemented a tool along the same lines to explore the World Wide Web [69]) anticipate the user’s future interaction by adding not only a new node to a frame, but also its immediate neighbors. Huang et al.

[68] or North [99] also include a control over throwing away some part of the logical frame to avoid saturation on the screen.

As far as the repositioning is concerned, the simplest solution is to use the same layout algorithm for each logical frame. This is done, for example, by Huang et al. [68]. (Note that the latter use a modified spring algorithm. This is one case where the relatively small graph on the screen makes the use of a force-directed method perfectly feasible in graph visualization.) North [99] and Brandes and Wagner [14] go further by providing dynamic control over the parameters that direct the layout algorithms.

As said above, this line of visual graph management is still quite new, but we think that it will gain in importance in the years to come and that it will complement the navigation and exploration methods described elsewhere in this survey.

## 4 CLUSTERING

As mentioned earlier, it is often advantageous to reduce the number of visible elements being viewed. Limiting the number of visual elements to be displayed both improves the clarity and simultaneously increases performance of layout and rendering [78]. Various “abstraction” and “reduction” techniques have been applied by researchers in order to reduce the visual complexity of a graph. One approach is to perform clustering.

Clustering is the process of discovering groupings or classes in data based on a chosen semantics. Clustering techniques have been referred to in the literature as *cluster analysis, grouping, clumping, classification, and unsupervised pattern recognition* [41], [89]. We will refer to clustering that uses only structural information about the graph as *structure-based clustering* (also referred to as identifying *natural clusters* [109]). The use of the semantic data associated with the graph elements to perform clustering could be termed *content-based clustering*.

Although content-based clustering can yield groupings which are most appropriate for a particular application and can even be combined with structure-based clustering, most mentions of clustering in graph visualization are references to purely structure-based clustering, with a few notable exceptions [91], [105]. This is probably due to the fact that content-based clustering requires application-specific data and knowledge. Any application which implements content-based clustering is likely to be so specialized to a problem domain that it is no longer general enough for use in other application areas. Furthermore, an advantage of using structure-based clustering is that natural clusters often retain the structure of the original graph, which can be useful for user orientation in the graph itself.

It is important to note that clustering can be used to accomplish functions such as filtering and search. In visualization terms, filtering usually refers to the deemphasis or removal of elements from the view, while search usually refers to the emphasis of an element or group of elements. Both filtering and search can be accomplished by partitioning elements into two or more groups and, then, emphasizing one of the groups.

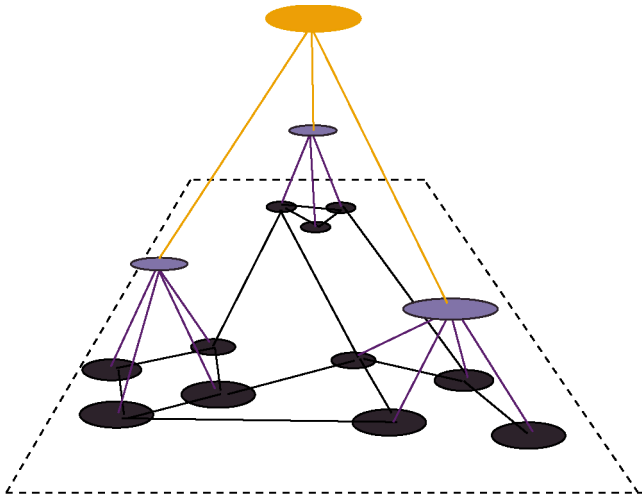


Fig. 21. A structure induced by hierarchical clustering. (Adapted from Eades and Feng [37].)

By far the most common clustering approach in graph visualization is to find clusters which are disjoint or mutually exclusive, as opposed to clusters that overlap (found by a process called *clumping*). Disjoint clusters are simpler to navigate than overlapping clusters because a visit of the clusters only visits the members once. It should be noted, however, that it is not always possible to find disjoint clusters, such as in the case of language-oriented or semantic topologies.

A common technique for finding natural clusters is to choose the clustering with the least number of edges between members. This technique is described by Mirkin [89]. It is also known as the Ratio Cut technique in VLSI design [124]. This technique extends to the case when edges have a weight. The task is then to minimize the total weight of the edges connecting members [109]. Natural clusters can also be obtained by applying a spring model (see below).

#### 4.1 Layout of a Clustered Graph

After discovering clusters within the data, we can reduce the number of elements to display by restricting our view to the clusters themselves. This provides an overview of the structure and allows us to retain a context while reducing visual complexity. Looking at the simpler and smaller clustered graph, the user should be better able to grasp the overall structure of the graph. Most algorithms look for a balance between the number of clusters and the number of nodes within clusters [1], [31]. A small number of clusters allows for fast processing and navigation. However, this number should not be too small because, otherwise, the visible information content is too low.

A common technique is to represent the clusters with glyphs and treat them as super-nodes in a higher-level or *compound graph*, which we can now navigate instead of the original graph. Some approaches have already been proposed [37], [112]. Huang and Eades [70] also give a precise definition of how edges between super-nodes can be induced (they refer to this idea as *abridgement*). This technique has also been implicitly implemented in many other visualization systems. One original solution is to omit the edges and position the nodes in a way that indicates

their connectivity [126]. This solution eliminates the problem of edge-crossings and reduces visual clutter.

If clustering is performed by successively applying the same clustering process to groups discovered by a previous clustering operation, the process is referred to as *hierarchical clustering* [89]. A containment hierarchy will result from hierarchical clustering and this may be navigated as a tree, with each cluster represented as a node in the tree (see Fig. 21). Hierarchical clustering can therefore be used to induce a hierarchy in a graph structure that might not otherwise have a hierarchical structure.

The approaches discussed until now involve first finding logical clusters, then laying out the graph of clusters. A completely different approach to clustering is based on force-directed layout. It lets forces between nodes influence the position of the node in the layout. All nodes in the system exert repulsive force on the others and related nodes are attracted to each other. After several iterations in which the positions are adjusted according to the calculated force, the system stabilizes, yielding clusters which are visually apparent. In a case study of Narcissus [60], the authors report that this technique can produce useful clusters in a relatively small number of iterations. As with other N-body problems, the complexity is  $O(N^3)$ . Another example of clustering by layout is described for the SemNet system [42], where clustering is accomplished by using semantic information to determine the positioning of nodes.

#### 4.2 Node Metrics for Clustering

In order to cluster a graph, we must use numerical measures associated with the nodes. A node metric can be used to measure or to quantify an abstract feature associated with a node in order to compare it with others of the same type and acquire a ranking. A metric can be implemented as a numeric computable function. Clustering can be accomplished by assigning elements to groups according to their metric value. Metrics can also be used to implement search or filtering in which elements with a certain metric value or a value above a threshold are highlighted.

The term *metric*, or *node metric*, has been used in many different ways in graph visualization. In this survey, we will use the term to refer to *a measure that is associated with a node in the graph*. We have identified the concept of node metrics in several places in the literature [11], [50], [61], [78]. Of course, similar concepts can be applied to metrics associated with edges.

A metric is structure-based if it only uses information about the structure of the graph. A metric is content-based if it uses information or data associated with the node such as text. The advantage of a structural metric is that no domain knowledge is required. This makes a structural metric useful for all applications. It is possible, of course, to combine structural and content-based metrics for more powerful effects. A simple approach is to allow the user to add an application-specific "weight" to the nodes, which is then combined with the structural metric [50], [61], [62].

An example of a structural metric is the degree of a node (i.e., the number of edges connected to the node). With such a metric, the application could exclusively display the nodes with a degree higher than or equal to a

threshold value. This would give a view of data which shows the nodes that have the largest number of relations with other nodes. A metric more specific to trees (called the Strahler metric [120]) has been applied in Fig. 22, in which nodes with the highest Strahler metric values generate a *skeleton* or backbone which is then emphasized (see Herman et al. [61], [62]).

Metrics can also be composed due to their numeric nature [62]. By choosing, for example, the weighted average of metrics, the user can choose how much influence a particular feature has on the resulting composed metric and thereby influence the resulting clustering. The *Degree of Interest (DOI)* function of Furnas [50] is also an example of a metric that is composed of two other metrics (in this case, a metric based on distance and a level of detail).

Node metrics can be used for many different purposes and, in our view, all the possible applications have not yet been fully explored. For instance, metrics can also be used to govern a spanning tree extraction procedure (see Section 2.3). Furnas's DOI function has been used to generate a focus+context view of the graph.<sup>11</sup> In another application, metrics are used to influence layout [127].

Once a subset of nodes has been selected, as with a skeleton, a method of representing the unselected nodes must be chosen. In the case of clustering, the selected set of nodes is the set of super-nodes or the groups themselves. Kimelman et al. name three possible approaches [78] (see Fig. 22):

- *Ghosting*: deemphasizing nodes, or relegating nodes to the background.
- *Hiding*: simply not displaying the unselected nodes. This is also referred to as folding or eliding.
- *Grouping*: grouping nodes under a new super-node representation.

These approaches may be combined, for example, with clusters represented by transparent super-nodes used by Sprenger et al. [116] in the IVORY system. Fig. 22c demonstrates an alternative where the size and the shape of the glyph representing the grouping is used to indicate the structure of the underlying subgraph. The resulting graph, technically a compound graph, is a sort of high-level map or *schematic view* [23], [62] of the original graph, which is useful for navigation of the original graph.

Clustering is full of challenges and is applied in many different fields, which has the unfortunate consequence that results about clustering are disseminated in journals and conferences addressing very different topics. This makes it difficult to gather the results into a unified theory or into a structured set of methodologies. Surprisingly, the book by Battista et al. [5] does not include a chapter on clustering, although the Graph Drawing Symposia welcomes papers on the topic every year. Our feeling is that this issue should receive more attention in the future, especially from the information visualization community.

11. As mentioned earlier, although Furnas referred to this technique as "fish-eye," his technique is not limited to fish-eye in the geometric sense, as described in Section 3.2.1.

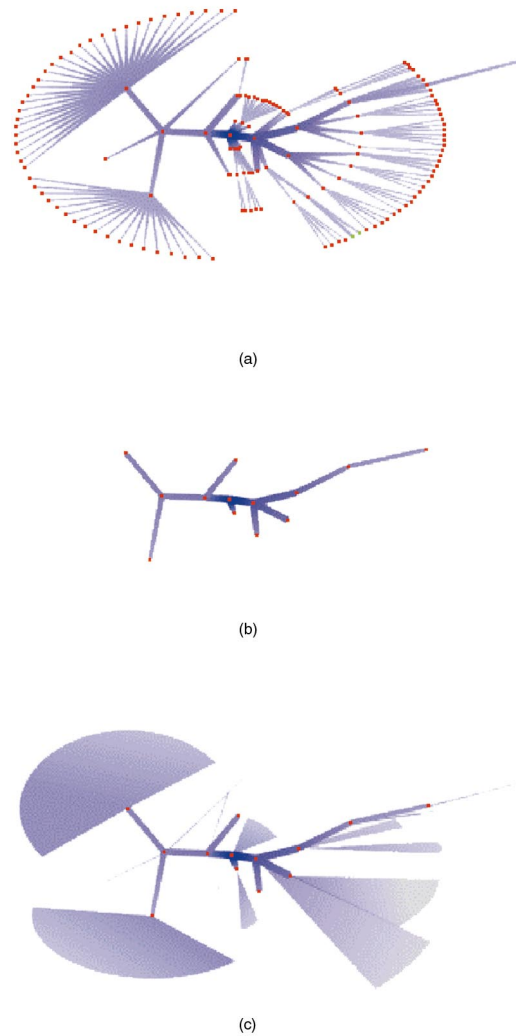


Fig. 22. Different schematic views of a tree: (a) ghosting, (b) hiding, and (c) grouping.

## 5 SYSTEMS

The area of graph visualization has reached a level of maturity in which large applications and application frameworks are being developed. However, it is difficult to enumerate all the systems because of the sheer quantity. Furthermore, some of them have a short lifetime because they are research tools and others are embedded in specialized applications. An overview of all graph visualization systems would go beyond the scope of this survey. However, we have already referred to a number of systems in earlier sections, based on features that we found interesting or important. Some other systems also caught our attention. Without any claim to completeness, we briefly describe a few additional systems below.

Efforts to develop software libraries and frameworks have been underway in several places. Some libraries are directed at mathematicians and include large libraries of algorithms, while others are meant for more general application. Some of the libraries and frameworks that are available are GTL [45], LINK [8], GFC [21], GDT [55], and GVF [64]. Although there is no widely used standard for

graph description formats, GML [66] and GraphXML [65] are available.

SemNet [42] is one of the few systems to provide graph editing while still providing a comprehensive set of tools to visualize large graphs. It is also one of the earliest complete systems that we know about.

Clustering has been applied by many older systems such as SemNet [42], Narcissus [60], SKETCH [118], and the Navigational View Builder [91]. Some newer systems that cluster graphs are NicheWorks [126], DA-TU [70], STARLIGHT [105], and a new system used by Bell Laboratories [58] for network visualization.

NicheWorks is an example of a complete system implementation that can be adapted for very specific applications. As an example, it has been used to visualize Y2K related problems [39]. The *fsviz* system of Carrière and Kazman [20], the da Vinci system of the University of Bremen [48], or the Latour system developed at CWI [63] fall into the same category. We should also mention the company called Tom Sawyer Software,<sup>12</sup> which offers a number of products based on various graph drawing techniques.

A few systems stand out because of unique features. The STARLIGHT [105] system performs content-based clustering and allows multiple mappings and layouts. It is one of the few systems that allows a 3D graph to be mapped to locations on a plane (for associating nodes or entire graphs with geographical positions). Shiozawa et al. [114] use a similar type of 3D to 2D mapping in order to view cell dependencies in a spreadsheet application. Another system, SDM [24] is unique because of a method of filtering in which nodes of interest are selected from a cityscape view by a plane above them. A similar cityscape view of nodes is used by Chen and Carr [22]. A system called WebPath [46] uses a fog effect in a 3D rendering of web history to limit the window of viewing. Graphs have also been used in an attempt to understand images and the transformations on them, where edges represent operations [85]. A system for viewing Bayesian Belief Networks [129] is one of a unique few (including [8], [63]) to employ animation for informative purposes. A highly interactive system called Constellation [95] has sophisticated zooming and highlighting features that facilitate the analysis of linguistic networks.

The World Wide Web is one of the typical application areas where graph visualization may play an important role in the future. H3View [93], based on hyperbolic viewing (see Section 2.5), is part of a Web site management tool of SGI, whereas the similar ideas of Lamping et al. [82], [83] are also exploited by a commercial spin-off of Xerox, called Inxight.<sup>13</sup> Earlier in this survey, we referred to NESTOR [1] or WebOFDAV [69], which can be used as web navigation tools. Other examples in this category are the Harmony Browser [1], Mapa [32], or Fetuccino [7] (the latter also combines the results of a web search engine with graph visualization).

12. <http://www.tomsawyer.com>.

13. <http://www.inxight.com>.

## 6 JOURNALS AND CONFERENCES

This survey is based on an extensive literature overview drawn from various conferences and journals. One of the difficulties of the field is that results are spread over a large number of different publications. To help the reader in pursuing research in the area, we list here some of the main publications which may be of interest:

- The *Graph Drawing Symposia* are organized yearly at various locations in the World. The proceedings are published by Springer-Verlag. These symposia have evolved into the traditional meeting places of the graph drawing community.
- The new *Journal of Graph Algorithms and Applications* (JGAA) is an on-line journal which gathers a similar community as the graph drawing symposia. The home page of the journal is at Brown University,<sup>14</sup> but Oxford University Press will also publish the collected papers in book formats.
- Graph drawing has strong relationships with computational geometry and algorithms. As a consequence, specialized journals like *Computational Geometry: Theory and Applications* or *Algorithmica* might also be a valuable source, although the papers in these journals tend to be much more “mathematical”, hence, more difficult to read for the computer graphics and information visualization communities.
- As said before, the yearly *CHI’XX* and *UIST’XX* conferences, both sponsored by ACM SIGCHI, often contain important papers for information visualization due to the importance of the user interface issue. Similarly, the *ACM Transactions on Human Computer Interaction* can be a valuable source of information.
- The yearly *InfoViz’XX* symposia form a separate track within the well-known *IEEE Visualization* conference. These symposia, as well as the Visualization conference itself, have become one of the leading events in the area by now.
- Somewhat confusingly, there is also a yearly *IEEE Conference on Information Visualization* which, however, has no real connection to the *InfoViz’XX* symposia (besides being sponsored by IEEE, too). Our own experience is that the academic level of *InfoViz’XX* is somewhat better.
- What was known before as the series of *Eurographics Workshop on Scientific Computing’XX* has recently changed its name to *Data Visualization’XX*, with information visualization as a separate track. The workshops have become joint Eurographics IEEE TCVG symposia and are considered as the European “sister” conference to IEEE Visualization.
- Some traditional computer graphics journals, like the *IEEE Transactions on Visualization and Computer Graphics* or the *Computer Graphics Forum* (which include the proceedings of the Eurographics conferences, too), have an increasing number of papers in information visualization.

14. <http://www.cs.brown.edu/publications/jgaa>.



- Finally, application-oriented journals or conference proceedings may also include papers on information visualization related to their respective application area. Examples include the proceedings of the yearly *XXth World Wide Web* or the *Digital Library'XX* conferences.

Obviously, the list is not exhaustive, but, hopefully, it is still useful for the reader as a starting point.

## REFERENCES

- [1] C.J. Alpert and A.B. Kahng, "Recent Developments in Netlist Partitioning: A Survey," *Integration: The VLSI J.*, vol. 19, pp. 1-81, 1995.
- [2] K. Andrews, "Visualizing Cyberspace: Information Visualization in the Harmony Internet Browser," *Proc. IEEE Symp. Information Visualization (InfoViz '95)*, pp. 97-105, 1995.
- [3] P.K. Argawal, B. Aronov, J. Pach, R. Pollack, and M. Sharir, "Quasi-Planar Graphs Have a Linear Number of Edges," *Proc. Symp. Graph Drawing, GD '95*, pp. 1-7, 1995.
- [4] G. di Battista, P. Eades, R. Tamassia, and I.G. Tollis, "Algorithms for Drawing Graphs: An Annotated Bibliography," *Computational Geometry: Theory and Applications*, vol. 4, no. 5, pp. 235-282, 1994.
- [5] G. di Battista, P. Eades, R. Tamassia, and I.G. Tollis, *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1999.
- [6] R.A. Becker, S.G. Eick, and A.R. Wilks, "Visualizing Network Data," *IEEE Trans. Visualization and Computer Graphics*, vol. 1, no. 1, pp. 16-28, 1995.
- [7] I. Ben-Shaul, M. Herscovici, M. Jacovi, Y.S. Maarek, D. Pelleg, M. Shtalham, V. Soroka, and S. Ur, "Adding Support for Dynamic and Focused Search with Fetuccino," *Proc. Eighth Int'l World Wide Web Conf.*, pp. 575-587, 1999.
- [8] J. Berry, N. Dean, M. Goldberg, G. Shannon, and S. Skiena, "Graph Drawing and Manipulation with LINK," *Proc. Symp. Graph Drawing GD '97*, pp. 425-437, 1999.
- [9] F. Bertault, "A Force-Directed Algorithm that Preserves Edge Crossing Properties," *Proc. Symp. Graph Drawing, GD '99*, pp. 351-358, 1999.
- [10] J. Blythe, C. McGrah, and D. Krackhardt, "The Effect of Graph Layout on Inference from Social Network Data," *Proc. Symp. Graph Drawing, GD '95*, pp. 40-51, 1995.
- [11] R.A. Botafogo, E. Rivlin, and B. Schneiderman, "Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics," *ACM Trans. Information Systems*, vol. 10, no. 2, 1992.
- [12] F.J. Brandenburg, M. Himsolt, and C. Rohrer, "An Experimental Comparison of Force-Directed and Randomized Graph Drawing Algorithms," *Proc. Symp. Graph Drawing GD '95*, 1996.
- [13] U. Brandes, G. Shubina, and R. Tamassia, "Improving Angular Resolution in Visualizations of Geographic Networks," *Data Visualization '2000, Proc. Joint Eurographics and IEEE TCVG Symp. Visualization*, to appear.
- [14] U. Brandes and D. Wagner, "A Bayesian Paradigm for Dynamic Graph Layout," *Proc. Symp. Graph Drawing GD '97*, pp. 236-247, 1997.
- [15] S.K. Card, G.G. Robertson, and W. York, "The WebBook and the Web Forager: An Information Workspace for the World Wide Web," *Human Factors in Computer Systems, CHI '96 Conf. Proc.*, pp. 111-117, 1996.
- [16] *Readings in Information Visualization*, S.K. Card, J.D. Mackinlay, and B. Shneiderman, eds. Morgan Kaufmann, 1999.
- [17] M.S.T. Carpendale, D.J. Cowperthwaite, and F.D. Fracchia, "3D Pliable Surfaces," *Proc. UIST '95 Symp.*, pp. 217-266, 1995.
- [18] M.S.T. Carpendale, D.J. Cowperthwaite, F.D. Fracchia, and T. Shermer, "Graph Folding: Extending Detail and Context Viewing into a Tool for Subgraph Comparisons," *Proc. Symp. Graph Drawing GD '95*, pp. 127-139, 1996.
- [19] M.S.T. Carpendale, D.J. Cowperthwaite, and F.D. Fracchia, "Extending Distortion Viewing from 2D to 3D," *IEEE Computer Graphics and Applications*, vol. 17, no. 4, pp. 42-51, 1997.
- [20] J. Carrière and R. Kazman, "Research Report: Interacting with Huge Hierarchies: Beyond Cone Trees," *Proc. IEEE Conf. Information Visualization '95*, pp. 74-81, 1995.
- [21] C.L. Cesar, *Graph Foundation Classes for Java*. IBM, <http://www.alphaWorks.ibm.com/tech/gfc>, 1999.
- [22] C. Chen and L. Carr, "Visualizing the Evolution of a Subject Domain: A Case Study," *Proc. IEEE Visualization '99 Conf.*, pp. 449-452, 1999.
- [23] M.C. Chuah, "Dynamic Aggregation with Circular Visual Designs," *Proc. IEEE Symp. Information Visualization (InfoViz '98)*, pp. 30-37, 1998.
- [24] M.C. Chuah, S.F. Roth, J. Mattis, and J. Kolojechick, "SDM: Malleable Information Graphics," *Proc. IEEE Symp. Information Visualization*, pp. 36-42, 1995.
- [25] H.S.M. Coxeter, *Introduction to Geometry*. John Wiley & Sons, 1973.
- [26] I.F. Cruz and R. Tamassia, "Online Tutorial on Graph Drawing," <http://www.cs.brown.edu/people/rt/papers/gd-tutorial/gd-constraints.pdf>. year?
- [27] I.F. Cruz and J.P. Twarog, "3D Graph Drawing with Simulated Annealing," *Proc. Symp. Graph Drawing GD '95*, pp. 162-165, 1995.
- [28] R. Davidson and D. Harel, "Drawing Graphs Nicely Using Simulated Annealing," *ACM Trans. Graphics*, vol. 15, no. 4, pp. 301-331, 1996.
- [29] E. Dengler and W. Cowan, "Human Perception of Laid-Out Graphs," *Proc. Symp. Graph Drawing GD '98*, pp. 441-444, 1998.
- [30] A. Denise, M. Vasconcellos, and D.J.A. Welsh, "The Random Planar Graph," *Congressus Numerantium*, vol. 113, pp. 61-79, 1996.
- [31] C.A. Duncan, M.T. Goodrich, and S.G. Kobourov, "Balanced Aspect Trees and Their Use for Drawing Very Large Graphs," *Proc. Symp. Graph Drawing GD '98*, pp. 111-124, 1998.
- [32] D. Durand and P. Kahn, "MAPA," *Proc. Ninth ACM Conf. Hypertext and Hypermedia (Hypertext '98)*, 1998.
- [33] P. Eades, "A Heuristic for Graph Drawing," *Congressus Numerantium*, vol. 42, pp. 149-160, 1984.
- [34] P. Eades and K. Sugiyama, "How to Draw a Directed Graph," *J. Information Processing*, vol. 13, no. 4, pp. 424-434, 1990.
- [35] P. Eades, "Drawing Free Trees," *Bulletin of the Inst. for Combinatorics and Its Applications*, pp. 10-36, 1992.
- [36] P. Eades and S.H. Whitesides, "Drawing Graphs in Two Layers," *Theoretical Computer Science*, vol. 131, no. 2, pp. 361-374, 1994.
- [37] P. Eades and Q.-W. Feng, "Multilevel Visualization of Clustered Graphs," *Proc. Symp. Graph Drawing GD '96*, pp. 101-112, 1997.
- [38] P. Eades, M.E. Houle, and R. Webber, "Finding the Best Viewpoints for Three-Dimensional Graph Drawings," *Proc. Symp. Graph Drawing GD '97*, pp. 87-98, 1998.
- [39] S.G. Eick, "A Visualization Tool for Y2K," *Computer*, vol. 31, no. 10, pp. 63-69, 1998.
- [40] J. Eklund, J. Sawers, and R. Zeiliger, "NESTOR Navigator: A Tool for the Collaborative Construction of Knowledge through Constructive Navigation," *Proc. Ausweb '99, Fifth Australian World Wide Web Conf.*, 1999.
- [41] B. Everitt, *Cluster Analysis*, first ed. Heinemann Educational Books Ltd., 1974.
- [42] K.M. Fairchild, S.E. Poltrock, G.W. Furnas, "SemNet: Three-Dimensional Representation of Large Knowledge Bases," *Cognitive Science and Its Applications for Human-Computer Interaction*, pp. 201-233, Lawrence Erlbaum Assoc., 1988.
- [43] K.M. Fairchild, "Information Management Using Virtual Reality-Based Visualisations," *Virtual Reality: Application and Explorations*, Academic Press, 1993.
- [44] A. Formella and J. Keller, "Generalized Fisheye Views of Graphs," *Proc. Symp. Graph Drawing GD '95*, pp. 242-253, 1995.
- [45] M. Forster, A. Pick, and M. Raitner, *Graph Template Library*, Univ. of Passau, <http://infosun.fmi.uni-passau.de/GTL/>, 1999.
- [46] E. Frécon and G. Smith, "WebPath—A Three Dimensional Web History," *Proc. IEEE Symp. Information Visualization (InfoViz '98)*, 1998.
- [47] A. Frick, A. Ludwig, and H. Mehldau, "A Fast Adaptive Layout Algorithm for Undirected Graphs," *Proc. Symp. Graph Drawing GD '93*, pp. 389-403, 1994.
- [48] M. Fröhlich and M. Werner, "Demonstration of the Interactive Graph Visualization System da Vinci," *Proc. DIMACS Workshop Graph Drawing '94*, 1995.
- [49] T.M.J. Fruchterman and E.M. Reingold, "Graph Drawing by Force-Directed Placement," *Software—Practice & Experience*, vol. 21, pp. 1,129-1,164, 1991.
- [50] G.W. Furnas, "Generalized Fisheye Views," *Human Factors in Computing Systems, CHI '86 Conf. Proc.*, pp. 16-23, 1986.

- [51] G.W. Furnas and B.B. Bederson, "Space-Scale Diagrams: Understanding Multiscale Interfaces," *Human Factors in Computing Systems, CHI '95 Conf. Proc.*, pp. 234-241, 1995.
- [52] G.W. Furnas and X. Zhang, "MuSE: A Multi-Scale Editor," *Proc. UIST '98 Symp.*, 1998.
- [53] M.R. Garey and D.S. Johnson, "Crossing Number is NP-Complete," *SIAM J. Algebraic and Discrete Methods*, vol. 4, no. 3, pp. 312-316, 1983.
- [54] A. Garg and R. Tamassia, "On the Computational Complexity of Upward and Rectilinear Planarity Testing," *Proc. Symp. Graph Drawing, GD '95*, pp. 286-297, 1995.
- [55] *Graph Drawing Toolkit*. Third Univ. of Rome, <http://www.dia.uniroma3.it/~gdt/>, 1999.
- [56] C. Gunn, "Visualizing Hyperbolic Space," *Proc. Eurographics Workshop Computer Graphics and Math.*, pp. 299-313, 1992.
- [57] B. Hausmann, B. Slopanka, and H.-P. Seidel, "Exploring Plane Hyperbolic Geometry," *Proc. Workshop Visualization and Math.*, pp. 21-36, 1998.
- [58] T. He, "Internet-Based Front-End to Network Simulator," *Data Visualization '99, Proc. Joint Eurographics and IEEE TCVG Symp. Visualization*, pp. 247-252, 1999.
- [59] M. Hemmje, C. Kunkel, and A. Willet, "LyberWorld—A Visualization User Interface Supporting Fulltext Retrieval," *Proc. ACM SIGIR '94*, 1994.
- [60] R.J. Hendley, N.S. Drew, A.M. Wood, and R. Beale, "Narcissus: Visualising Information," *Proc. IEEE Symp. Information Visualization*, pp. 90-96, 1995.
- [61] I. Herman, M. Delest, and G. Melançon, "Tree Visualization and Navigation Clues for Information Visualization," *Computer Graphics Forum*, vol. 17, no. 2, pp. 153-165, 1998.
- [62] I. Herman, M.S. Marshall, G. Melançon, D.J. Duke, M. Delest, and J.-P. Domenger, "Skeletal Images as Visual Cues in Graphs Visualization," *Data Visualization '99, Proc. Joint Eurographics and IEEE TCVG Symp. Visualization*, pp. 13-22, 1999.
- [63] I. Herman, G. Melançon, M.M. de Ruitter, and M. Delest, "Latour—A Tree Visualization System," *Proc. Symp. Graph Drawing GD '99*, pp. 392-399, 1999. A more detailed version in: *Reports of the Centre for Math. and Computer Sciences*, Report number INS-R9904, available at: <http://www.cwi.nl/InfoVisu/papers/LatourOverview.pdf>, 1999.
- [64] I. Herman, M.S. Marshall, and G. Melançon, "An Object-Oriented Design for Graph Visualization," *Reports of the Centre for Math. and Computer Sciences*, Report no. INS-R0001, available at: <http://www.cwi.nl/InfoVisu/GVF/GVF.pdf>, 2000.
- [65] I. Herman and M.S. Marshall, "GraphXML," *Reports of the Centre for Math. and Computer Sciences*, available at: <http://www.cwi.nl/InfoVisu/GVF/GraphXML/GraphXML.pdf>, 1999.
- [66] M. Himsolt, *GML—Graph Modelling Language*, Univ. of Passau, <http://infosun.fmi.uni-passau.de/Graphlet/GML/>, 1997.
- [67] J. Hopcroft and R.E. Tarjan, "Efficient Planarity Testing," *J. ACM*, vol. 21, no. 4, pp. 549-568, 1974.
- [68] M.L. Huang, P. Eades, and J. Wang, "Online Animated Graph Drawing Using a Modified Spring Algorithm," *J. Visual Languages and Computing*, vol. 9, no. 6, 1998.
- [69] M.L. Huang, P. Eades, and R.F. Cohen, "WebOFDAV—Navigating and Visualizing the Web On-Line with Animated Context Swapping," *Proc. Seventh World Wide Web Conf.*, pp. 636-638, 1998.
- [70] M.L. Huang and P. Eades, "A Fully Animated Interactive System for Clustering and Navigating Huge Graphs," *Proc. Symp. Graph Drawing GD '98*, pp. 374-383, 1998.
- [71] C.-S. Jeong and A. Pang, "Reconfigurable Disc Trees for Visualizing Large Hierarchical Information Space" *Proc. IEEE Symp. Information Visualization (InfoViz '98)*, 1998.
- [72] B. Johnson and B. Schneiderman, "Tree-Maps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures," *Proc. IEEE Visualization '91*, pp. 275-282, 1991.
- [73] M. Juenger and P. Mutzel, "2-Layer Straightline Crossing Minimization: Performance of Exact and Heuristic Algorithms," *J. Graph Algorithms and Applications*, vol. 1, pp. 33-59, 1997.
- [74] D. Jungnickel, *Graphs, Networks and Algorithms*. Springer Verlag, 1999.
- [75] T. Kamada and S. Kawai, "An Algorithm for Drawing General Undirected Graphs," *Information Processing Letters*, vol. 31, pp. 7-15, 1989.
- [76] K. Kaugars, J. Reinfelds, and A. Brazma, "A Simple Algorithm for Drawing Large Graphs on Small Screens," *Proc. Symp. Graph Drawing GD '94*, pp. 278-281, 1995.
- [77] T.A. Keahey and E.L. Robertson, "Techniques for Non-Linear Magnification Transformations," *Proc. IEEE Symp. Information Visualization (InfoViz '97)*, pp. 38-45, 1997.
- [78] D. Kimelman, B. Leban, T. Roth, and D. Zernik, "Reduction of Visual Complexity in Dynamic Graphs," *Proc. Symp. Graph Drawing GD '93*, 1994.
- [79] M.R. Laguna, R. Martí, and V. Vals, "Arc Crossing Minimization in Hierarchical Digraphs with Tabu Search," *Computers and Operations Research*, vol. 24, no. 12, pp. 1,165-1,186, 1997.
- [80] M. Laguna and R. Martí, "GRASP and Path Relinking for 2-Layer Straight Line Crossing Minimization," *INFORMS J. Computing*, vol. 11, pp. 44-52, 1999.
- [81] M. Laguna and R. Martí, "Heuristics and Meta-Heuristics for 2-Layer Straight Line Crossing Minimization," URL: <http://www-bus.colorado.edu/Faculty/Laguna/>, 1999.
- [82] J. Lamping, R. Rao, and P. Pirollo, "A Focus+context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies," *Human Factors in Computing Systems, CHI '95 Conf. Proc.*, 1995.
- [83] J. Lamping and R. Rao, "The Hyperbolic Browser: A Focus+context Technique for Visualizing Large Hierarchies," *J. Visual Languages and Computing*, vol. 7, no. 1, pp. 33-55, 1996.
- [84] Y.K. Leung and M.D. Apperly, "A Review and Taxonomy of Distortion-Oriented Presentation Techniques," *ACM Trans. Computer-Human Interaction*, vol. 1, no. 2, pp. 126-160, 1994.
- [85] K.L. Ma, "Image Graphs—A Novel Approach to Visual Data Exploration," *Proc. IEEE Visualization '99*, pp. 81-88, 1999.
- [86] M. McGrath, J. Blythe, and D. Krackhardt, "The Effect of Spatial Arrangement on Judgments and Errors in Interpreting Graphs," *Social Networks*, vol. 19, no. 3, pp. 223-242, 1997.
- [87] G. Melançon and I. Herman, "Circular Drawings of Rooted Trees," *Reports of the Centre for Math. and Computer Sciences*, report number INS-9817, available at: <http://www.cwi.nl/InfoVisu/papers/circular.pdf>, 1998.
- [88] K. Mehlhorn and P. Mutzel, "On the Embedding Phase of the Hopcroft and Tarjan Planarity Testing Algorithm," *Algorithmica*, vol. 16, pp. 233-242, 1996.
- [89] B. Mirkin, *Mathematical Classification and Clustering*. Kluwer Academic, 1996.
- [90] K. Misue, P. Eades, W. Lai, and K. Sugiyama, "Layout Adjustment and the Mental Map," *J. Visual Languages and Computing*, vol. 6, pp. 183-210, 1995.
- [91] S. Mukherjee, J.D. Foley, and S. Hudson, "Visualizing Complex Hypermedia Networks through Multiple Hierarchical Views," *Human Factors in Computing Systems, CHI '95 Conf. Proc.*, pp. 331-337, 1995.
- [92] T. Munzner and P. Burchard, "Visualizing the Structure of the World Wide Web in 3D Hyperbolic Space," *Proc. VRML '95 Symp.*, 1995.
- [93] T. Munzner, "H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space," *Proc. 1997 IEEE Symp. Information Visualization (InfoViz '97)*, pp. 2-10, 1997.
- [94] T. Munzner, "Drawing Large Graphs with H3Viewer and Site Manager," *Proc. Symp. Graph Drawing GD '98*, pp. 384-393, 1998.
- [95] T. Munzner, F. Guimbretière, and G. Robertson, "Constellation: A Visualization Tool for Linguistic Queries from MindNet," *Proc. IEEE Symp. Information, InfoVis '99*, pp. 132-135, 1999.
- [96] P. Mutzel, C. Gutwenger, R. Brockenaue, S. Fialko, G. Klau, M. Kruger, T. Ziegler, S. Naher, D. Alberts, D. Ambras, G. Koch, M. Junger, C. Buchein, and S. Leipert, "A Library of Algorithms for Graph Drawing," *Proc. Symp. Graph Drawing GD '97 Symp.*, pp. 456-457, 1998.
- [97] T. Munzner, E. Hoffman, K. Claffy, and B. Fenner, "Visualizing the Global Topology of the MBone," *Proc. IEEE Symp. Information Visualization*, 1996.
- [98] L. Nigay and F. Vernier, "Design Method of Interaction Techniques for Large Information Space," *Proc. Advanced Visual Interfaces (AVI '98)*, 1998.
- [99] S. North, "Incremental Layout in DynaDAG," *Proc. Symp. Graph Drawing GD '95*, pp. 409-418, 1995.
- [100] H.C. Purchase, "Which Aesthetic Has the Greatest Effect on Human Understanding?" *Proc. Symp. Graph Drawing GD '97*, pp. 248-261, 1998.
- [101] H.C. Purchase, R.F. Cohen, and M. James, "Validating Graph Drawing Aesthetics," *Proc. Symp. Graph Drawing GD '95*, pp. 435-446, 1995.

- [102] H.C. Purchase, R.F. Cohen, and M. James, "An Experimental Study of the Basis for Graph Drawing Algorithms," *ACM J. Experimental Algorithmics*, vol. 2, no. 4, 1997.
- [103] E.M. Reingold and J.S. Tilford, "Tidier Drawing of Trees," *IEEE Trans. Software Eng.*, vol. 7, no. 2, pp. 223-228, 1981.
- [104] J. Rekimoto and M. Green, "The Information Cube: Using Transparency in 3D Information Visualization," *Proc. Third Ann. Workshop Information Technologies & Systems (WITS '93)*, 1993.
- [105] J.S. Risch, D.B. Rex, S.T. Dowson, T.B. Walters, R.A. May, and B.D. Moon, "The STARLIGHT Information Visualization System," *Proc. IEEE Conf. Information Visualization*, pp. 42-49, 1997.
- [106] G.G. Robertson, J.D. Mackinlay, and S.K. Card, "Cone Trees: Animated 3D Visualizations of Hierarchical Information," *Human Factors in Computing Systems, CHI '91 Conf. Proc.*, pp. 189-194, 1991.
- [107] G.G. Robertson, S.K. Card, and J.D. Mackinlay, "Information Visualization Using 3D Interactive Animation," *Comm. ACM*, vol. 36, no. 4, pp. 57-71, 1993.
- [108] A. Robinson, *EBI Hyperbolic Viewer*. European Bioinformatics Inst., available at: <http://industry.ebi.ac.uk/~alan/components>, 1998.
- [109] T. Roxborough and A. Sen, "Graph Clustering Using Multiway Ratio Cut," *Proc. Symp. Graph Drawing GD '97*, pp. 291-296, 1998.
- [110] M. Sarkar and M.H. Brown, "Graphical Fish-Eye Views of Graphs," *Human Factors in Computing Systems, CHI '92 Conf. Proc.*, pp. 83-91, 1992.
- [111] M. Sarkar and M.H. Brown, "Graphical Fisheye Views," *Comm. ACM*, vol. 37, no. 12, pp. 73-84, 1994.
- [112] D. Schaffer, Z. Zuo, S. Greenberg, L. Bartram, J. Dill, S. Dubs, and M. Roseman, "Navigating Hierarchically Clustered Networks through Fisheye and Full-Zoom Methods," *ACM Trans. Computer-Human Interaction*, vol. 3, no. 2, pp. 162-188, 1996.
- [113] Y. Shiloach, "Arrangements of Planar Graphs on the Planar Lattices," PhD thesis, Weizmann Inst. of Science, Rehovot, Israel, 1976.
- [114] H. Shiozawa, K.-i. Okada, and Y. Matsushita, "3D Interactive Visualization for Inter-Cell Dependencies of Spreadsheets," *Proc. IEEE Symp. Information Visualization (InfoViz '99)*, pp. 79-82, 1999.
- [115] G. Sindre, B. Gulla, and H.G. Jokstad, "Onion Graphs: Aesthetics and Layout," *Proc. IEEE/CS Symp. Visual Languages (VL '93)*, pp. 287-291, 1993.
- [116] T.C. Sprenger, M. Gross, D. Bielser, and T. Strasser, "IVORY—An Object-Oriented Framework for Physics-Based Information Visualization in Java," *Proc. IEEE Symp. Information Visualization (InfoViz '98)*, 1998.
- [117] K. Sugiyama, S. Tagawa, and M. Toda, "Methods for Visual Understanding of Hierarchical Systems Structures," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 11, no. 2, pp. 109-125, 1989.
- [118] K. Sugiyama and K. Misue, "Visualization of Structural Information: Automatic Drawing of Compound Digraphs," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 4, pp. 876-892, 1991.
- [119] W. Tutte, "How to Draw a Graph," *Proc. London Math. Soc.*, vol. 3, no. 13, pp. 743-768, 1963.
- [120] X.G. Viennot, "Trees Everywhere," *Proc. 15th CAAP Conf.*, pp. 18-41, 1990.
- [121] J.Q. Walker II, "A Node-Positioning Algorithm for General Trees," *Software—Practice and Experience*, vol. 20, no. 7, pp. 685-705, 1990.
- [122] C. Ware and G. Franck, "Evaluation of Stereo and Motion Cues for Visualising Information in Three Dimensions," *ACM Trans. Graphics*, vol. 15, no. 2, pp. 121-140, 1996.
- [123] C. Ware, *Information Visualization: Perception for Design*. Morgan Kaufmann, 2000.
- [124] Y.C. Wei and C.K. Cheng, "Ratio Cut Partitioning for Hierarchical Designs," *IEEE Trans. Computer-Aided Design*, vol. 10, no. 7, pp. 911-921, 1991.
- [125] J.J. van Wijk and H. van de Wetering, "Cushion Treemaps: Visualization of Hierarchical Information," *Proc. IEEE Symp. Information Visualization (InfoViz '99)*, pp. 73-78, 1999.
- [126] G.J. Wills, "Niche Works—Interactive Visualization of Very Large Graphs," *Proc. Symp. Graph Drawing GD '97*, pp. 403-415, 1998.
- [127] R.M. Wilson and R.D. Bergeron, "Dynamic Hierarchy Specification and Visualization," *Proc. IEEE Symp. Information Visualization (InfoViz '99)*, pp. 65-72, 1999.
- [128] P. Young, "Three Dimensional Information Visualization (Survey)," Computer Science Technical Report, Centre for Software Maintenance Dept. of Computer Science, Univ. of Durham, available at: <http://www.dur.ac.uk/~dcs3py/pages/work/documents/lit-survey/IV-Survey/index.html>, 1996.

- [129] J.-D. Zapata-Rivera, E. Neufeld, and J.E. Greer, "Visualization of Bayesian Belief Networks," *Proc. IEEE Visualization '99, Late Breaking Hot Topics*, pp. 85-88, 1999.
- [130] R. Zeiliger, "Supporting Constructive Navigation of Web Space," *Proc. Workshop Personalized and Solid Navigation in Information Space*, 1998.



**Ivan Herman** graduated as applied mathematician in 1979 in Budapest, Hungary, and received his PhD at the University of Leiden, The Netherlands, in 1990. He is currently a senior researcher at the Centre for Mathematics and Computer Science (CWI) in Amsterdam and is head of the research group on information visualization. He has been chief designer and implementor of several graphics and multimedia systems, and is also author or coauthor of close to 50 scientific publications in international journals and conferences. He is currently cochair of the Ninth World Wide Web conference and of the second joint Eurographics/IEEE TSVG Symposium on Visualization. He has been a member of the Eurographics Executive Committee since 1987 and a member of its Executive Board since 1990. He is also member of the IEEE Computer Society and of the Advisory Committee of the World Wide Web Consortium.



**Guy Melançon** received his PhD in mathematics from the University of Québec in Montréal, Canada, in 1991 and recently defended his "habilitation" in computer science at the University of Bordeaux I, France. He is currently a scientific researcher at the Centre for Mathematics and Computer Science (CWI) in Amsterdam and also holds a permanent position at the University of Bordeaux I, France. He is the author or coauthor of many scientific publications in international journals and conferences in combinatorial mathematics and information visualization. He is currently coorganizer of the second joint Eurographics/IEEE TSVG Symposium on Visualization.



**M. Scott Marshall** received a BA in computer science from the University of California at Berkeley in 1992. He is currently a research associate at the Centre for Mathematics and Computer Science (CWI) in Amsterdam, working in the information visualization research group. He recently helped to implement an ISO standard for multimedia called PREMIO and coauthored a book on the subject. His research interests include scientific, medical, and information visualization and knowledge representation. He is currently working on his PhD dissertation on graph visualization in cooperation with the University of Bordeaux, France.

# Visualizing the Non-Visual: Spatial analysis and interaction with information from text documents

James A. Wise, James J. Thomas, Kelly Pennock, David Lantrip,  
Marc Pottier, Anne Schur, Vern Crow  
Pacific Northwest Laboratory  
Richland, Washington

## Abstract

*This paper describes an approach to IV that involves spatializing text content for enhanced visual browsing and analysis. The application arena is large text document corpora such as digital libraries, regulations and procedures, archived reports, etc. The basic idea is that text content from these sources may be transformed to a spatial representation that preserves informational characteristics from the documents. The spatial representation may then be visually browsed and analyzed in ways that avoid language processing and that reduce the analysts' mental workload. The result is an interaction with text that more nearly resembles perception and action with the natural world than with the abstractions of written language.*

## 1: Introduction

Information Visualization (IV), extends traditional scientific visualization of physical phenomena to diverse types of information (e.g. text, video, sound, or photos) from large heterogeneous data sources. It offers significant capability to different kinds of analysts who must identify, explore, discover, and develop understandings of complex situations.

IV has been studied for many centuries, integrating techniques from art and science in its approach [7, 9]. The information analyst's perspective illustrates that their process involves more than envisioning information. [4] It is both the visual representations and the resultant interactions with it that entail the analyst's work.

Current visualization approaches demonstrate effective methods for visualizing mostly structured and/or hierarchical information such as organization charts, directories, entity-attribute relationships, etc. [3,9]. Free text visualizations have remained relatively unexamined.

The idea that open text fields themselves or raw prose might be candidates for information visualization is not obvious. Some research in information retrieval utilized graph theory or figural displays as 'visual

query' tools on document bases [5,8], but the information returned is documents in their text form--which the user still must read to cognitively process. The need to read and assess large amounts of text that is retrieved through even the most efficient means puts a severe upper limit on the amount of text information that can be processed by any analyst for any purpose.

At the same time, "Open Source" digital information--the kind available freely or through subscription over the Internet--is increasing exponentially. Whether the purpose be market analysis, environmental assessment, law enforcement or intelligence for national security, the task is to peruse large amounts of text to detect and recognize informational 'patterns' and pattern irregularities across the various sources. But modern information technologies have made so much text available that it overwhelms the traditional reading methods of inspection, sift and synthesis.

## 2: Visualizing text

True text visualizations that would overcome these time and attentional constraints must represent textual content and meaning to the analyst without them having to read it in the manner that text normally requires. These visualizations would instead result from a content abstraction and spatialization of the original text document that transforms it into a new visual representation that communicates by image instead of prose. Then the image could be understood in much the way that we explore our worldly visual constructions.

It is thus reasonable to hypothesize that across the purposes for perusing text, some might be better satisfied by transforming the text information to a spatial representation which may then be accessed and explored by visual processes alone. For any reader, the rather slow serial process of mentally encoding a text document is the motivation for providing a way for them to instead use their primarily preattentive, parallel processing powers of visual perception.

The goal of text visualization, then, is to spatially transform text information into a new visual

representation that reveals thematic patterns and relationships between documents in a manner similar if not identical to the way the natural world is perceived. This is because the perceptual processes involved are the results of millions of years of selective mammalian and primate evolution, and have become biologically tuned to seeing in the natural world. The human eye has its own contrast and wavelength sensitivity functions. It has prewired retinal "textons", or primitive form elements used to quickly build up components of complex visual images. Much of this processing takes place in parallel on the retinal level, and so is relatively effortless, exceptionally fast, and not additive to cognitive workload. Even at the visual cortex, perception appears to rely on spatially distributed parallel construction processes in a topography that corresponds to the real physical world. The central conjecture behind the approach to text visualization described here is that the same spatial perceptual mechanisms that operate on the real world will respond to a synthetic one, if analogous cues are present and suitably integrated. The bottleneck in the human processing and understanding of information in large amounts of text can be overcome if the text is spatialized in a manner that takes advantage of common powers of perception.

### 3. Visualization transformations: from text to pictures

Four important technical considerations need to be addressed in the creation of useful visualizations from raw text. First, there must be a clear definition of what comprises text and how it can be distinguished from other symbolic representations of information. Second, there must be a way to transform raw text into a different visual form that retains much of the high dimensional invariants of natural language, yet better enables visual exploration and analysis by the individual. Third, suitable mathematical procedures and analytical measures must be defined as the foundation for meaningful visualizations. Finally, a database management system must be designed to store and manage text and all of its derivative forms of information.

For the purpose of this paper, text is a written alphabetical form of natural language. Diagrams, tables, and other symbolic representations of language are not considered text. Text has statistical and semantic attributes such as the frequency and context of individual words, and the combinations of words into topics or themes. The differences between text's statistical and semantic compositions provide much of

the opportunity for the text visualizations described in this paper. For example, reading a text document to extract its semantic meaning is different from learning that a document is of a certain relative size, type, or authorship, with particular content themes. But both semantic and content knowledge can be valuable to an analyst. Identifying publishing activity on particular subjects from particular authors at certain places and times is useful, especially if one does not have to read all of the documents to determine that pattern.

In digital form, written text can be treated statistically to extract information about its content and context, if not semantic meaning. While this does not necessarily entail natural language processing algorithms, it does require a set of special purpose processes to convert text to an alternative spatial form that can be displayed and utilized without needing to read it.

The first component of a software architecture to visualize text is the document database or corpora. Documents contained within such databases are derived from messages, news articles, regulations, etc., but contain primarily textual material. The next component is the text processing engine, which transforms natural language from the document database to spatial data. The output from the text engine is either stored directly in a visualization database, or projected onto a low dimensional, visual representation. Other components of the architecture are the Graphical User Interface (GUI), the display software (such as visualization packages), the Applications Interface (API), and auxiliary tools.

#### 3.1: Processing text

The primary requirements of a text processing engine for information visualization are: 1) the identification and extraction of essential descriptors or text features, 2) the efficient and flexible representation of documents in terms of these text features, and 3) subsequent support for information retrieval and visualization.

Text features are typically one of three general types, though any number of variations and hybrids are possible. The first type is frequency-based measures on words, utilizing only first order statistics. The presence and count of unique words in a document identifies those words as a feature set. The second type of feature is based on higher order statistics taken on the words or letter strings. Here, the occurrence, frequency, and context of individual words are used to characterize a set of explicit or implicitly (e.g. associations defined by a neural

network) defined word classes. The third type of text feature is semantic in nature. The association between words is not defined through analysis of the word corpus, as with statistical features, but is defined a priori using knowledge of the language. Semantic approaches may utilize natural or quasi-natural language understanding algorithms, so that the semantic relationships (i.e., higher-order information) are obtained.

Text features are a "shorthand" representation of the original document, satisfying the need of a text engine to be an efficient and flexible representation of textual information. Instead of a complex and unwieldy string of words, feature sets become the efficient basis of document representations and manipulations. The feature set information must be complete enough to permit flexible use of these alternatives. Text engines support both efficiency and flexibility, though these criteria are often in opposition.

The third requirement of the text engine is to support information retrieval and visualization. The text processing engine must provide easy, intuitive access to the information contained within the corpus of documents. Information retrieval implies a query mechanism to support it. This may include a basic Boolean search, a high level query language, or the visual manipulation of spatialized text objects in a display. To provide efficient retrieval, the text processing engine must pre-process documents and efficiently implement an indexing scheme for individual words or letter strings.

The more visual aspect of information retrieval is known as information browsing. The specificity of querying has a counterpoint in the generality of browsing. The text processing engine or subsidiary algorithms can support browsing by providing composite or global measures which produce an intuitive index into topics or themes contained within the text corpus. A set of measures which characterize the text in meaningful ways provide for multiple perspectives of documents and their relationships to one another. One example of such a measure is "similarity". Based on the occurrence and the context of key words or other extracted features, measures of similarity can be computed which reflect the relatedness between documents. When similarity is represented as spatial proximity or congruity of form, it is easily visualized. A diversity of measures is essential, given that documents can be extraordinarily complex entities containing a large number of imprecise topics and subtopics. Clearly, no single visualizable snapshot of a document base can provide the whole picture.

### 3.2: Visualizing output from text processing

Composing a spatial representation from the output of the text analysis engine is the next step to visualizing textual information. Spatialization itself is composed of several stages. The first involves representing the document, typically as a vector in a high dimensional feature space. The vector representation is the initial spatial expression of the document, and a variety of comparisons, filters, and transformations can be made from it directly.

To represent each of these documents, an initial visualization may consist of a scatter plot of points (one for each document), collocated according to a measure of similarity based on vector representations. Since visualization of the textual information requires a low-dimensional representation of documents that inhabit a high-dimensional space, projection is necessary. Typically, linear or non-linear Principal Components Analysis or metric Multi-Dimensional Scaling (MDS) can be used to reduce dimensionality to a visualizable subspace. One serious concern with these techniques, is their exponential order of complexity, requiring that dimensionality reduction and scaling be considered simultaneously since a large corpus may contain 20,000 or more documents. For large document corpora, alternatives to the projection of each document point are necessary. In these cases, clustering can be performed in the high-dimensional feature space and the cluster centroids become the objects to be visualized. For a review of clustering and metric issues, see [11].

### 3.3: Managing the representation

There are two basic classes of data that must be managed. The first is the raw text files for each document. The text itself as well as a variety of header information fall into this category. This first class of data is static in nature, simple in structure, and therefore easy to manage. The second broad class of data is the visual forms of the text. This class of data is derived from the numerous algorithms designed to cluster, structure, and visually present information, and is both extensive and dynamic.

For the current text visualization endeavor, an object-oriented database was selected for managing text and its various visual forms. This paradigm was chosen for its flexibility of data representation, the power of inheritance, and the ease of data access where complexity of the data structures to be managed is great. The structures contain both high and low dimensional spaces, substructures such as clusters and

super clusters, entities such as documents and cluster centroids, and a variety of other components. The class structure of the database also permits the common elements to be shared (inherited) through the hierarchy of data classes, while the differences between the structures can be specified at lower levels. The selected object-oriented database also implements database entities as persistent objects, where the access and manipulation of the data are one and the same, eliminating the need for a query mechanism as such.

### 3.4: Interface design for text visualization

To achieve direct engagement for text visualization [6], the interface must provide 1) a preconscious visual form for information 2) interactions which sustain and enrich the process of knowledge building, 3) a fluid environment for reflective cognition and higher-order thought, and 4) a framework for temporal knowledge building.

Three primary display types are made available to the analyst. Tools are arranged along the perimeter of the display monitor and can be used as operators on the representations. Conversely, the representations or selected areas in the representations can be dragged and dropped onto the tools to spawn the appropriate action. The analyst can work on the primary information views [1] in an area known as the backdrop, which serves as a central display resource for visual information; alternatively, she can move the views to the workshop or the chronicle. The workshop is a grid where selected views or parts of views can be placed for work and/or visual review. The grid has a number of resizable windows to hold multiple views. The chronicle is a space where representations of more enduring interest can be placed. Views placed in this area can be linked to form a sequenced visual story where decision points are highlighted. The workshop and the chronicle take advantage of the phenomena of visual momentum; the ability to extract information across a set of successively viewed displays [10] that can be a series of static or dynamic images. The characteristics of the backdrop, workshop, and chronicle, known collectively as storylining, provide the ability to capture and visually organize situations across the time-past, present, and future. This endows the analyst with the ability to summarize their experience of knowledge building [2].

## 4: Examples from the MVAB Project

The Multidimensional Visualization and Advanced Browsing project is currently exploring a number of

representations for the visualization and analysis of textual information. These approaches have been showcased in SPIRE™, the Spatial Paradigm for Information Retrieval and Exploration, which was developed to facilitate the browsing and selection of documents from large corpora (20,000+ documents). Described below are the two major visualization approaches or views which were developed in the first year of this project: Galaxies and Themescapes.

Starfields and topographical maps were selected as display metaphors because they offer a rich variety of cognitive spatial affordances that naturally address the problems of text visualization. Starfields create point clusters which suggest patterns of interest. Maps offer topographies of peaks and valley that can be easily detected based on contour patterns. Both these spatial arrangements allow overview and detail without a change of view. Each view, however, offers a different perspective of the same information and serves as the organizing points for knowledge construction.

### 4.1: Galaxies

The Galaxies visualization displays cluster and document interrelatedness by reducing a high dimensional representation of documents and clusters to a 2D scatterplot of 'docupoints' that appear as do stars in the night sky. Although the resulting visualization is simple, it provides a critical first cut at sifting information and determining how the contents of a document base are related. The key measurement for understanding this visualization is the notion of document "similarity". The more similar that clusters and documents are to one another in terms of their context and content, the closer or more proximate they are located in the 2D space. By exploring and animating this visualization, analysts can quickly gain an understanding of patterns and trends that underlie the documents within a corpus. At the highest level of representation, Galaxies displays corpus clusters and the gisting terms which describe them. (Figure 1)

A simple glance at this spatial representation reveals the fundamental topics found within the corpus, and provides an avenue of exploration which can be followed by simply clicking on a cluster of interest to reveal the documents within. These documents can then be grouped, gisted, annotated, or retrieved for more detailed analysis. In addition to simple point and click exploration of the document base, a number of sophisticated tools exist to facilitate more in-depth analysis. An example of such a tool is the temporal slicer. Designed to help tie document

spatial patterns with temporal ones, this tool utilizes document timestamps to partition the document base into temporal units. The granularity of these units can be defined by the user as years, months, days, hours or minutes. Slicing a database entails moving a "temporal window" through the documents, and watching the visualization populate itself with documents. (Figure 2)

The resulting emergence of clusters can indicate temporal links that relate topics. When viewed in terms of known historical events and trends, these growing cluster patterns can provide insight into external causal relationships mirrored in the corpus.

#### 4.2: ThemeScapes

ThemeScapes are abstract, three-dimensional landscapes of information that are constructed from document corpora (Figure 3). The complex surfaces are intended to convey relevant information about

topics or themes found within the corpus, without the cognitive load encountered in reading such content. A thematic terrain simultaneously communicates both the primary themes of an arbitrarily large collection of documents and a measure of their relative prevalence in the corpus. Spatial relationships exhibited in the landscape reveal the intricate interconnection of themes, the transformation of themes across the whole of the document corpus, and the existence of information gaps, or "negative information."

The ThemeScapes' visual representation has several advantages. First, a ThemeScape displays much of the complex content of a document database. Elevation depicts theme strength, while other features of the terrain map such as valleys, peaks, cliffs, and ranges represent detailed interrelationships among documents and their composite themes. At a glance, it provides a visual thematic summary of the whole corpus. The second major advantage of thematic terrain is that it utilizes innate human abilities for

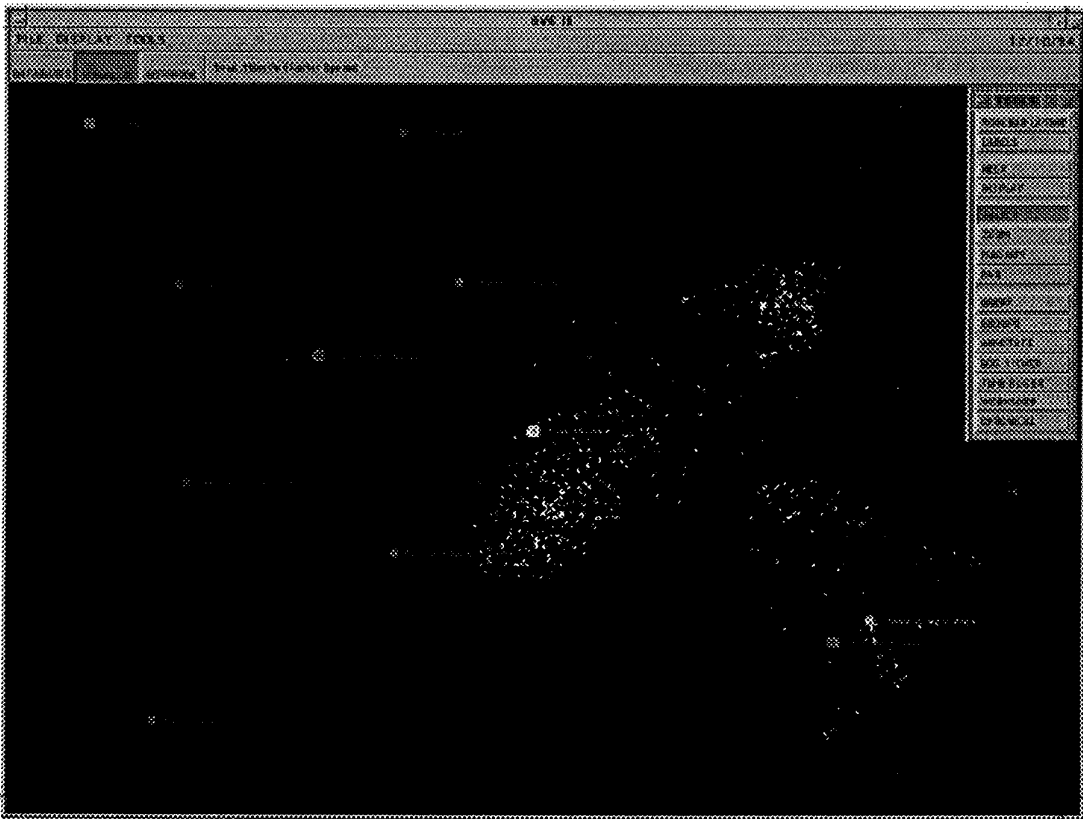
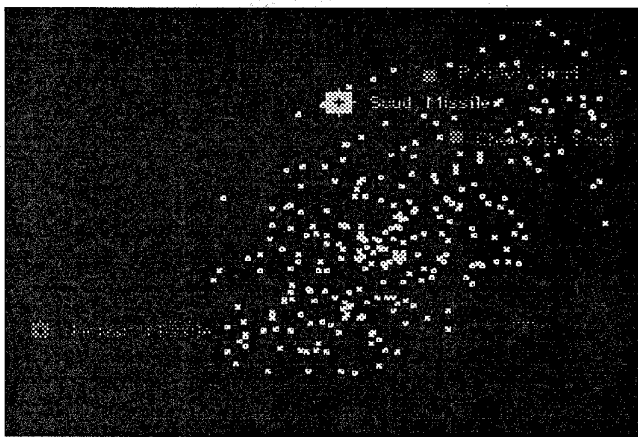
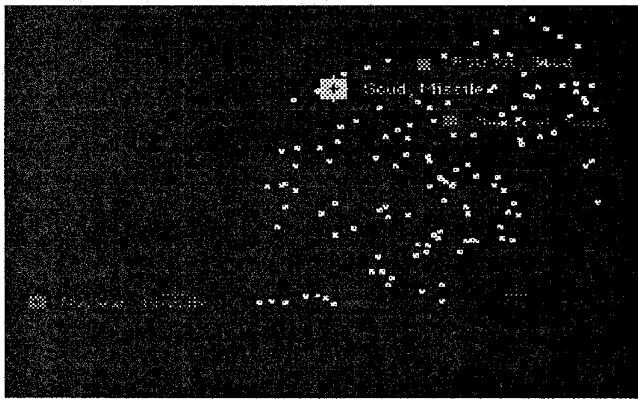
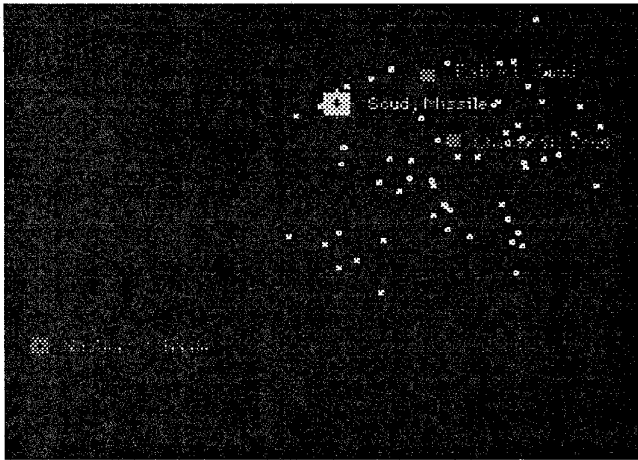


Figure 1: Galaxies visualization of documents and document clusters in a text database.





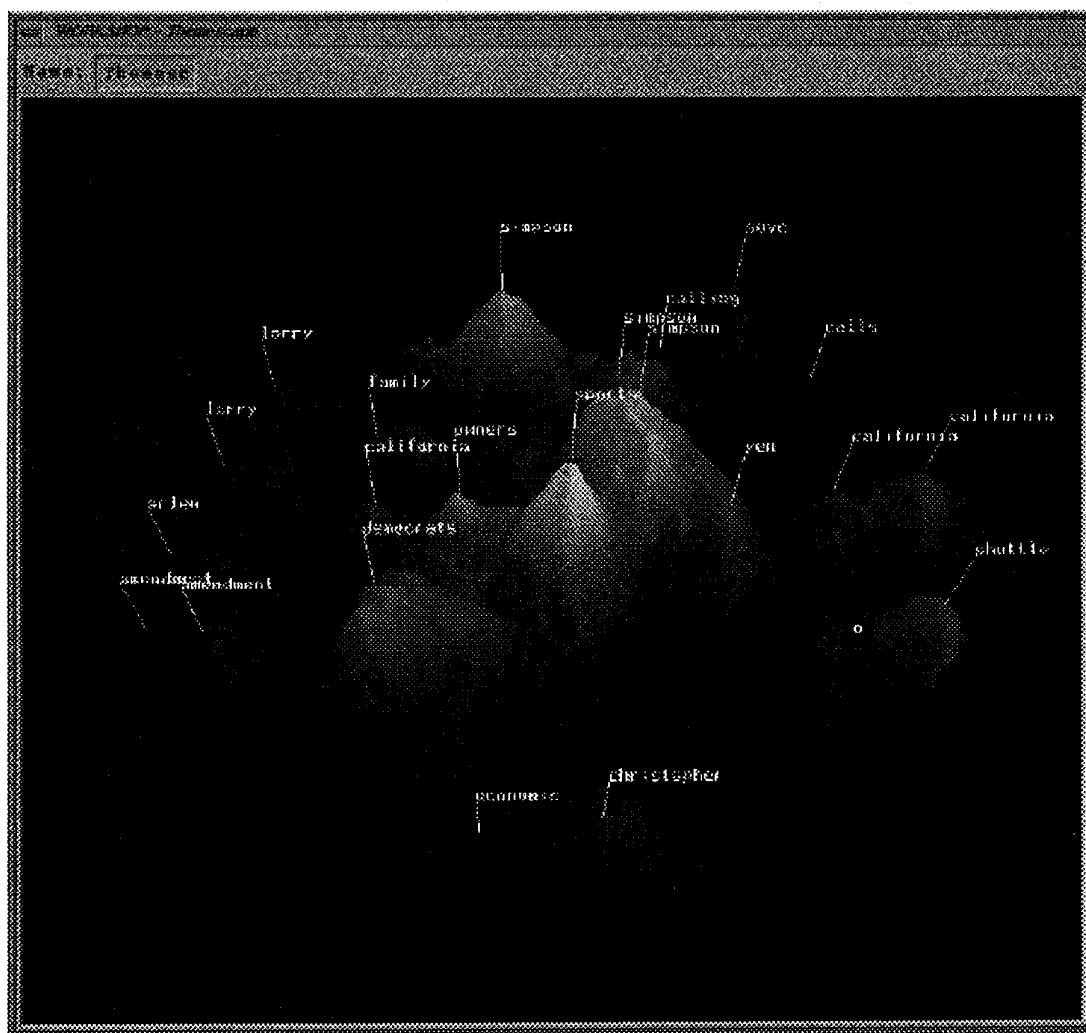
**Figure 2a, b, c: Users can slice a corpus to relate document patterns with temporal ones.**

pattern recognition and spatial reasoning. The complexity of the terrain is perceived and analyzed with parallel and preattentive processing which do not tax serial, attentional resources. This greatly expands the bandwidth of communication between the tool and the user. A third major advantage of the terrain implementation is its communicative invariance across levels of textual scale. An entire document corpus, a cluster of documents, individual documents, or even document components such as paragraphs or sentences can all be equally well visualized in a ThemeScape. This feature allows the ThemeScape to be used for automated document summarization as well as summarization of the whole document base, explicitly displaying the multitude of topics in a single image. Finally, ThemeScapes promote analysis by promoting exploration of the document space. Utilizing the metaphor of the landscape, associated tools allow the analyst to take 'core samples' and 'slices' through the thematic terrain to see its composition and to understand how thematic topics come to relate to one another in the underlying documents.

## **5: Conclusions and directions for future research and development**

The MVAB project has started with a visualization (Galaxies) that provides a simplified universal view of the relationships among documents in an entire corpus. We have then proceeded to the Themescape visualization, which does the same for the thematic content expressed in those documents. In doing so, we have gone from a metaphor of points in space to one of a landscape. We are now pursuing development of a third visualization that would handle specific entity-attribute relationships found in the documents, such as treated by Link Analysis. This layering of informational detail and abstraction appears necessary for large document bases where investigating global structure is a primary concern yet relationships between individual objects are also important. It gives real meaning and form to the notion of 'data mining'.

So far, the R &D efforts on the MVAB project have shown that there appears to be substantial justification to the idea that text visualizations can overcome much of the user limitations that results from accessing and trying to read from large document bases. Even with the relatively simple first Galaxies visualization of documents as stars in a 2-D space, analysts have returned reports of enhanced insight and time savings such as "discovering in 35 minutes what would have taken two weeks otherwise." Analysts



**Figure 3: A ThemeScape of an entire week of CNN newstories comprising a document corpus**

have also been quite creative with the tool, using the time slicer to do pattern recognition and comparison on evolving and historically documented situations. This kind of adaptation of what is essentially a visual browser has encouraged us to pursue analytical visualizations and to drop some distinctions among analysts' tasks that exist in the purely textual realm. For example, querying a document base by means of a Boolean text string is different from reading the documents returned from the query. However, the ThemeScape described above, although meant for visual analysis, also permits a different kind of visual querying than was ever possible under the all-or-none

choices of Boolean logic. Touching a ThemeScape can be a way to initiate a weighted query in terms of the themes that proportionally compose the elevation of the terrain. This allows the analyst to seek documents that talk about combinations of topics in a selected relative abundance based on the analyst's interests. Querying and analytical manipulation come together in a single visualization.

Users' experience also justifies the initial conviction that text visualizations will have to access and utilize the cognitive and visual processes that enable our spatial interactions with the natural world. This suggests visual metaphors that recapitulate

experiences of viewing the night sky and traversing landscapes.

Another observation echoed in the growing popularity of Visual Data Analysis (VDA) programs is that perception and action are provocative complements to one another. An image must be acted on in some way, which in turn suggests new facets of its character that stimulate further visual inspection. Galaxies success with analysts is in no small part due to the abilities to pan, group and timeslice the docupoints in the display. The success of other text visualizations will likely be determined by whether the user can manipulate them along the lines of their analytical intuitions.

Future efforts will elaborate the visual metaphors described above, as well as new ones that effectively capture how concepts and decisions 'come into form'. Much of the analyst's world is a dynamic changing information terrain. Seeking coherence and patterns in this environment carries a high price in time and effort. Capturing the development of a story or the threads of a concept communicated in prose is a high order for text visualizations. But there appears to be no formal reason why at least some of these aspects cannot be captured as well in image as they can in words.

Other extensions of this research are suggested by the addition of sensory modalities like sound to the text visualization. If text content or connections can be captured in three dimensional solid forms, then those forms might also be given other properties, like density, that characterize their appearance and behavior in the real world. Through enhanced means of 'virtual interaction', these properties could reinforce and extend the impressions gained by visual inspection alone, and start to give much more of the affective content and tone that well written prose conveys.

It is evident that the potentials of text visualization are just beginning to be explored and realized. With them, the incredible diversity and volumes of written information available around the world may yet be made more accessible and comprehensible through this perceptual restructuring. And the limitations of an Information Age will not be set by the speed with which a human mind can read.

## References

- [1] Bannon, L. J., and Bodker, S., *Beyond the Interface: Encountering Artifacts in Use*. In Carroll J. M. (Ed.) *Designing Interaction: Psychology at the Human Computer Interface* pages 227-253. Cambridge, Cambridge University Press, 1991.
- [2] Henniger, S., Belkin, N., *Interfaces Issues and Iteration Strategies for Information Retrieval Systems*. ACM Computer Interaction Tutorial Workbook #19, April 1994.
- [3] Johnson, J. A., Nardi, B. A., Zamer, C. L., and Miller, J. R., 1993. Information Visualization Using 3D Interactive Animation. *Communications of the ACM*, 36(4):40-56.
- [4] Keller, P. R., and M. M. Keller. *Visual Cues: Practical Data Visualization*. IEEE Computer Society Press, Los Alamitos, California. 1993.
- [5] Korfhage, Robert R. To See, or Not to See--Is That the Query? *Communications of the ACM*, 34, pages 134-141, 1991.
- [6] Laurel, B. *Computers as Theatre*. Addison-Wesley, Reading, Massachusetts, 1993.
- [7] Robertson, G. C., Card, S. K., and Mackinlay, J.D. 1993. Information Visualization Using 3D Interactive Animation. *Communications of the ACM*, 36(4):56-72
- [8] Spoerri, Anselm. InfoCrystal: A visual tool for information retrieval. *Proceedings of Visualization '93*, pages 150-157. IEEE Computer Society Press, Los Alamitos, California, 1993.
- [9] Tufte, E. R. *Envisioning Information*. Graphics Press, Cheshire, Connecticut, 1990.
- [10] Woods, D. D., *Visual Momentum: a Concept to Improve Cognitive Coupling of Person and Computer*. *International Journal of Man-Machine Studies* 21: 229-244. 1984.
- [11] York, J. and Bohn, S. *Clustering and Dimensionality Reduction in SPIRE*. Presented at the Automated Intelligence Processing and Analysis Symposium, Mar 28-30, 1995, Tysons Corner, VA.

# TileBars: Visualization of Term Distribution Information in Full Text Information Access

Marti A. Hearst

Xerox Palo Alto Research Center  
3333 Coyote Hill Rd, Palo Alto, CA 94304  
(415) 812-4742; hearst@parc.xerox.com

## ABSTRACT

The field of information retrieval has traditionally focused on textbases consisting of titles and abstracts. As a consequence, many underlying assumptions must be altered for retrieval from full-length text collections. This paper argues for making use of text structure when retrieving from full text documents, and presents a visualization paradigm, called TileBars, that demonstrates the usefulness of explicit term distribution information in Boolean-type queries. TileBars simultaneously and compactly indicate relative document length, query term frequency, and query term distribution. The patterns in a column of TileBars can be quickly scanned and deciphered, aiding users in making judgments about the potential relevance of the retrieved documents.

**KEYWORDS:** Information retrieval, Full-length text, Visualization.

## INTRODUCTION

Information access systems have traditionally focused on retrieval of documents consisting of titles and abstracts. As a consequence, the underlying assumptions of such systems are not necessarily appropriate for full text documents, which are becoming available online in ever-increasing quantities. Context and structure should play an important role in information access from full text document collections. A critical structural aspect of a full-length text is the pattern of distributions of the terms that comprise it. When a system retrieves a document in response to a query, it is important to indicate not only how strong the match is (e.g., how many terms from the query are present in the document), but also how frequent each term is, how each term is distributed in the text and where the terms overlap within the document. This information is especially important in long texts, since it is less clear how the terms in the query contribute to the ranking of a long text than a short abstract. The need for this kind

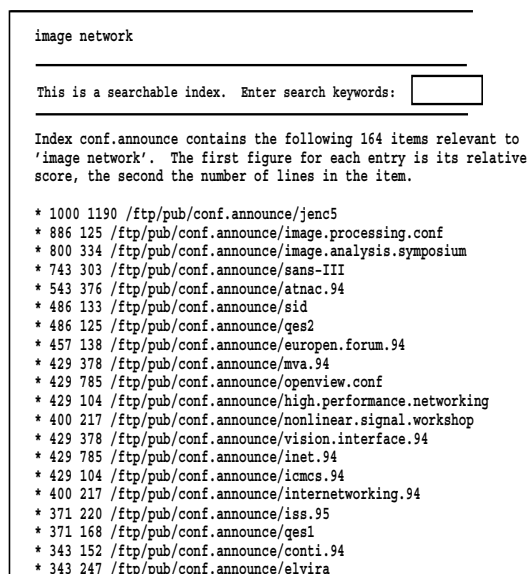


Figure 1: A sketch of the results of a WAIS search on *image* and *network* on a dataset of conference announcements.

of distributional information has not been emphasized in the past, perhaps in part because researchers had not focused on long texts.

To address these issues, I introduce a new display paradigm called *TileBars* which allows users to simultaneously view the relative length of the retrieved documents, the relative frequency of the query terms, and their distributional properties with respect to the document and each other. *TileBars* seem to be a useful analytical tool for understanding the results of Boolean-type queries, and preliminary work indicates they are useful for determining document relevance when applied to sample queries from a standard full-text test collection. This approach to visualization of the role of the query terms within the retrieved documents may also help explain why standard information retrieval measures succeed or fail for a given query.

## BACKGROUND: STANDARD INFORMATION RETRIEVAL

The purpose of information retrieval is to help users effectively access large collections of objects with the goal of satis-

fyng the users' stated information needs [6].<sup>1</sup> The most common approaches to text retrieval are Boolean term specification and similarity search. I use the term "similarity search" as an umbrella term covering the vector space model [26], probabilistic models [5], [12] and any other approach which attempts to find the documents that are most similar to a query or to one another based solely or primarily on the terms they contain.

Similarity search, in effect, ranks documents according to how close, in a multidimensional term space, combinations of the documents' terms are to combinations of the terms in the query. The closer two documents are to one another in the term space, the more topics they are presumed to have in common. This is a reasonable framework when comparing short documents, since the goal is often to discover which pairs of documents are most alike. For example, a query against a set of medical abstracts which contains terms for the name of a disease, its symptoms, and possible treatments is best matched against an abstract with as similar a constitution as possible. In similarity search, the best overall matches are not necessarily the ones in which the largest percentage of the query terms are found, however. For example, given a query of T terms, the vector space model permits a document that contains only a subset S of the query terms to be ranked relatively high if these terms occur infrequently in the corpus as a whole but frequently in the document.

In Boolean retrieval a query is stated in terms of disjunctions, conjunctions, and negations among sets of documents that contain particular words and phrases. Documents are retrieved whose contents satisfy the conditions of the Boolean statement. The users can have more control over what terms actually appear in the retrieved documents than they do with similarity search. In its basic form, Boolean search does not produce a ranking order, although ranking criteria as used in similarity search are often applied to the results of the Boolean search [11].

### The Problem with Ranking

There is great concern in the information retrieval literature about how to rank the results of Boolean and similarity searches. I contend that this concern is misplaced. Once a manageable subset of the thousands of available documents has been found, then the issue becomes a matter of providing the user with information that is informative and compact enough that it can be interpreted swiftly.<sup>2</sup> As discussed in the next subsection, there are many different ways in which a long text can be "similar" to the query that issued it, and so

<sup>1</sup>This paper will focus on collections of textual information only, although other media types apply as well.

<sup>2</sup>As further evidence for this viewpoint, Noreault et al. [23] performed an experiment on bibliographic records in which they tried every combination of 37 weighting formulas working in conjunction with 64 combining formulas on Boolean queries. They found that the choice of scheme made almost no difference: the best combinations got about 20% better than random ordering, and no one scheme stood out above the rest. These results imply that small changes to weighting formulas don't have much of an effect.

a system should supply the user with a way to understand the relationship between the retrieved documents and the query.

Furthermore, the standard approach to document ranking is opaque; users are unable to see what role their query terms played in the ranking of the retrieved documents. An ordered list of titles and probabilities is under-informative. The link between the query terms, the similarity comparison, and the contents of the texts in the dataset is too underspecified to assume that a single indicator of relevance can be assigned.

Instead, the representation of the retrieval results should present as many attributes of the texts and their relationship to the queries as possible, and present the information in a compact, coherent and accurate manner. Accurate in this case means a true reflection of the relationship between the query and the documents.

Consider for example what happens when one performs a keyword search using WAIS [18]. If the search completes, it results in a list of document titles and relevance rankings. The rankings are based on the query terms in some capacity, but it is unclear what role the terms play or what the reasons behind the rankings are. The length of the document is indicated by a number, which although interpretable, is not easily read from the display. Figure 1 represents the results of a search on *image* and *network* on a database of conference announcements. The user cannot determine to what extent either term is discussed in the document or what role the terms play with respect to one another. If the user prefers a dense discussion of images and would be happy with only a tangential reference to networking, there is no way to express this preference.

Attempts to place this kind of expressiveness into keyword based system are usually flawed in that the users find it difficult to guess how to weight the terms. If the guess is off by a little they may miss documents that might be relevant, especially because the role the weights play in the computation is far from transparent. Furthermore, the user may be willing to look at documents that are not extremely focused on one term, so long as the references to the other terms are more than passing ones. Finally, the specification of such information is complicated and time-consuming.

### The Importance of Document Structure

A problem with applying similarity search to full-length text documents is that the structure of full text is quite different from that of abstracts. Abstracts are compact and information-dense. Most of the (uncommon) terms in an abstract are salient for retrieval purposes because they act as placeholders for multiple occurrences of those terms in the original text, and because generally these terms pertain to the most important topics in the text. Consequently, if the text is of any sizeable length, it will contain many subtopic discussions that are never mentioned in its abstract, if one exists. On the other hand, an expository text may be viewed as a sequence of subtopics set against a "backdrop" of one or

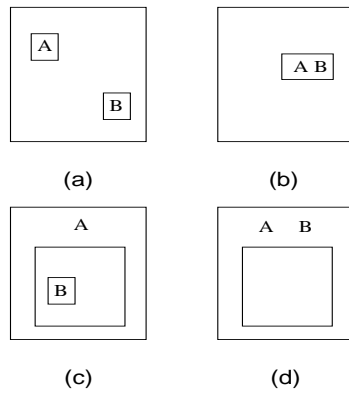


Figure 2: Possible relationships between two terms in a full text. (a) The distribution is disjoint, (b) co-occurring locally, (c) term A is discussed globally throughout the text, B is only discussed locally, (d) both A and B are discussed globally throughout the text.

two main topics. A long text is often comprised of many different subtopics which may be related to one another and to the backdrop in many different ways. The main topics of a text are discussed in its abstract, if one exists, but subtopics usually are not mentioned. Therefore, instead of querying against the entire content of a document, a user should be able to issue a query about a coherent subpart, or subtopic, of a full-length document, and that subtopic should be specifiable with respect to the document's main topic(s).

Figure 2 illustrates some of the possible distributional relationships between two terms in the main topic/subtopic framework. An information access system should be aware of each of the possible relationships and make judgments as to relevance based in part on this information. Thus a document with a main topic of "cold fusion" and a subtopic of "funding" would be recognizable even if the two terms do not overlap perfectly. The reverse situation would be recognized as well: documents with a main topic of "funding policies" with subtopics on "cold fusion" should exhibit similar characteristics.

The idea of the main topic/subtopic dichotomy can be generalized as follows: different distributions of term occurrences have different semantics; that is, they imply different things about the role of the terms in the text. The possible distribution relations that can hold between two sets of terms, and predictions about the usefulness of each distribution type, are enumerated and explained in [14].

### TextTiling: Automatic Discovery of Document Structure

To determine the kind of document structure described above, I have developed an algorithm, called *TextTiling*, that partitions expository texts into multi-paragraph segments that reflect their subtopic structure [15]. (Since the segments are adjacent and non-overlapping, they are called TextTiles.) The algorithm detects subtopic boundaries by analyzing the term

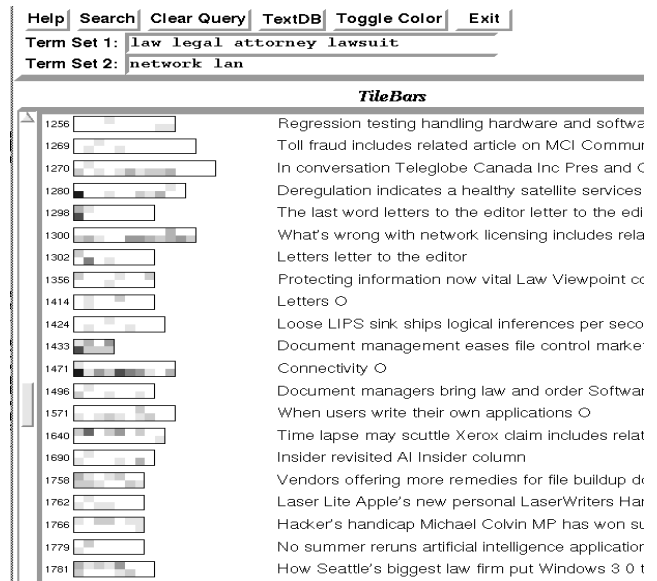


Figure 3: The TileBar display paradigm. Rectangles correspond to documents, squares correspond to text segments, the darkness of a square indicates the frequency of terms in the segment from the corresponding Term Set. Titles and the initial words of a document appear next to its TileBar.

repetition patterns within the text. The main idea is that terms that describe a subtopic will co-occur locally, and a switch to a new subtopic will be signalled by the ending of co-occurrence of one set of terms and the beginning of the co-occurrence of a different set of terms. In texts in which this assumption is valid, the central problem is determining where one set of terms ends and the next begins. The algorithm is domain-independent, and is fully implemented. The results of TextTiling are difficult to evaluate; comparisons to human judgments show the results are imperfect, as is often the case in fuzzy natural language processing tasks, but serviceable for their application to the task described below.

### TILEBARS

This section presents one solution to the problems described in the previous subsections. The approach is synthesized in reaction to three hypotheses:

- Long texts differ from abstracts and short texts in that, along with term frequency, term distribution information is important for determining relevance.
- The relationship between the retrieved documents and the terms of the query should be presented to the user in a compact, coherent, and accurate manner (as opposed to the single-point of information provided by a ranking).
- Passage-based retrieval should be set up to provide the user with the context in which the passage was retrieved, both within the document, and with respect to



Figure 4: TileBar search on (*patient medicine medical* AND *test scan cure diagnosis* AND *software program*) with some distribution constraints.

the query.

Figure 3 shows an example of a new representational paradigm, called TileBars, which provides a compact and informative iconic representation of the documents' contents with respect to the query terms. TileBars allow users to make informed decisions about not only which documents to view, but also which passages of those documents, based on the distributional behavior of the query terms in the documents. As mentioned above, the goal is to simultaneously indicate:

- (1) The relative length of the document,
- (2) The frequency of the term sets in the document, and
- (3) The distribution of the term sets with respect to the document and to each other.

Each large rectangle indicates a document, and each square within the document represents a TextTile. The darker the tile, the more frequent the term (white indicates 0, black indicates 8 or more instances, the frequencies of all the terms within a term set are added together). Since the bars for each set of query terms are lined up one next to the other, this produces a representation that simultaneously and compactly indicates relative document length, query term frequency, and query term distribution. The representation exploits the natural pattern-recognition capabilities of the human perceptual system [21]; the patterns in a column of TileBars can be quickly scanned and deciphered.

Term overlap and term distribution are both easy to compute and can be displayed in a manner in which both attributes together create easily recognized patterns. For example, overall darkness indicates a text in which both term sets are discussed

in detail. When both term sets are discussed simultaneously, their corresponding tiles blend together to cause a prominent block to appear. Scattered discussions have lightly colored tiles and large areas of white space.

TileBars make use of the following visualization properties (extracted from [27]):

- A variation in position, size, value [gray scale saturation], or texture is ordered [ordinal] that is, it imposes an order which is universal and immediately perceptible. [3]
- If shading is used, make sure differences in shading line up with the values being represented. The lightest (“unfilled”) regions represent “less”, and darkest (“most filled”) regions represent “more”. [20]
- Because they do have a natural visual hierarchy, varying shades of gray show varying quantities better than color. [29]

Note that the stacking of the terms in the query-specification portion of the document is reflected in the stacking of the tiling information in the TileBar: the top row indicates the frequencies of terms from Term Set 1 and the bottom row corresponds to Term Set 2. Thus the issue of how to specify the keyterms becomes a matter of what information to request in the interface. There is an implicit OR among the terms within a term set and an implicit AND between the term sets. Retrieved documents must have at least K hits from each term set, where K is an adjustable parameter.

TileBars allow users to be aware of what part of the document they are about to view before viewing it. To see what the document is about overall, they can simply mouse-click on the part of the representation that symbolizes the beginning of the document. Alternatively, they may go directly to a segment in the middle of the text in which terms from both term sets overlap, knowing in advance how far down in the document the passage occurs.

The TileBar representation allows for grouping by distribution pattern. Each pattern type occupies its own window in the display and users can indicate preferences by virtue of which windows they use. Thus there is no single correct ranking strategy: in some cases the user might want documents in which the terms overlap throughout; in other cases isolated passages might be appropriate. A variation of the interface organizes the retrieval results according to the distribution pattern type.

### Networks and the Law

Figure 3 shows some of the TileBars produced for the query on the term sets (*law legal attorney lawsuit*) AND (*network lan*) on the ZIFF collection [13]. (ZIFF is comprised mainly of commercial computer news.) In response to this query

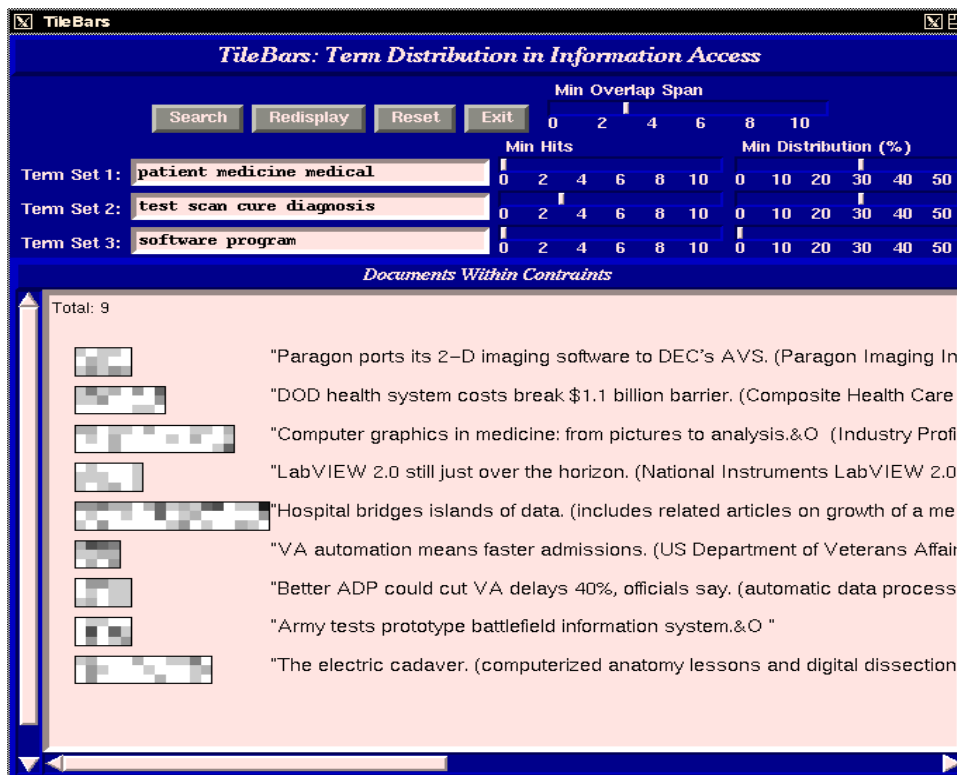


Figure 5: TileBar search on (*patient medicine medical AND test scan cure diagnosis AND software program*) with stricter distribution constraints.

one might expect documents about computer networks used in law firms, lawsuits involving illegal use of networks, and patent battles among network vendors. Since retrieval is on a collection of commercial computer texts, most instances of the word *network* will refer to the computer network sense, with exceptions for neural networks and perhaps some references to computer science theory and telephone systems. Since *legal* is an adjective, it can be used as a modifier in a variety of situations, but a strong showing of hits in its term set should indicate a legitimate legal discussion.

In the figure, the results have not been sorted in any manner other than document ID number. It is instructive to compare what the bars imply about the content of the texts with what actually appears in the texts. Document 1433 stands out because it appears to discuss both term sets in some detail. Documents 1300 and 1471 are also prominent because of a strong showing of the network term set. Document 1758 also has well-distributed instances of both term sets, although with less frequency than in document 1433. Legal terms have a strong distributional showing in 1640, 1766, 1781 as well. There are also several documents with very few occurrences of either term, although in some cases terms are more locally concentrated than in others. Most of the other documents look uninteresting due to their lack of overlap or infrequency of term occurrences.

Looking now at the actual documents we can determine the

accuracy of the inferences drawn from the TileBars. Clicking on the first tile of document 1433 brings up a window containing the contents of the document, centered on the first tile. The search terms are highlighted with two different colors, distinguished by term set membership, and the tile boundaries are indicated by ruled lines and tile numbers. The document describes in detail the use of a network within a legal office.

Looking at document 1300, the intersection between the term sets can be viewed directly by clicking on the appropriate tile. From the TileBar we know in advance that the tile to be shown appears about three quarters of the way through the document. Clicking here reveals a discussion of legal ramifications of licensing software when distributing it over the network. Document 1471 has only the barest instance of legal terms and so it is not expected to contain a discussion of interest – most likely a passing reference to an application. Indeed, the term is used as part of a hypothetical question in an advice column describing how to configure LANs. Note that a document like this would have been ranked highly by a mechanism that only takes into account term frequency.

The remaining documents with strong distributions of legal terms, 1758, 1640, 1766, 1781, discuss a documentation management system on a networked PC system in a legal office, a lawsuit between software providers, computer crime, and another discussion of a law firm using a new networked software system, respectively. Only the latter has overlap



with networking terms. Interestingly, the solitary mention of networking at the end of 1766 lists it as a computer crime problem to be worried about in the near future. This is an example of the suggestive nature of the positional information inherent in the representation.

Finally, looking at the seemingly isolated discussion of document 1298 we see a letter-to-the-editor about the lack of liability and property law in the area of computer networking. This letter is one of several letters-to-the-editor; hence its isolated nature. This is an example of a perhaps useful instance of isolated, but strongly overlapping, term occurrences. In this example, one might wonder why one legal term continues on into the next tile. This is a result of the tiling algorithm being slightly off in the boundary determination in this case.

As mentioned above, the remaining documents appear uninteresting since there is little overlap among the terms and within each tile the terms occur only once or twice. We can confirm this suspicion with a couple of examples. Document 1270 has one instance of a legal term; it is a passing reference to the former profession of an interview subject. Document 1356 discusses a court's legal decision about intellectual property rights on information. Tile 3 provides a list of ways to protect confidential information, one item of which is to avoid storing confidential information on a LAN. So in this case the reference to networks is only in passing.

Note that the conjunction of information about how much of each term set is present with how much the hits from each term set overlap provide indicate different kinds of information, which cannot be discerned from a ranking.

### Computer-aided Medical Diagnosis

Figures 4 and 5 show the results of a query on three term sets in a version of the interface that allows the user to restrict which documents are displayed according to several constraints: minimum number of hits for each term set, minimum distribution (the percentage of tiles containing at least one hit), and minimum adjacent overlap span. In this example the user is interested in documents that discuss computer-aided techniques for medical diagnosis, and the query is a conjunction of three term sets: (*patient medicine medical*) AND (*test scan cure diagnosis*) AND (*software program*). In Figure 4 the user has indicated that the document must contain a substantive discussion of the diagnosis terms, and that overlap among all three term sets must occur at least once within the span of three adjacent tiles. Note that this looser restriction yields some documents about computer-aided diagnosis with only passing references to medicine, which may indeed meet the user's information need. In figure 5, the user has emphasized the importance of the medical terms as well by specifying that displayed documents must have hits in at least 30% of their tiles. Judging from the titles displayed, this restriction was indeed useful in isolating documents of interest. Placing such constraints may cause relevant documents to be discarded, but an interface like this allows the user some

control over the ever-present tradeoff between showing only relevant documents and showing all relevant documents.

### Implementation Notes

The current implementation of the information access method underlying the TileBar display makes use of  $\approx 132,000$  documents of the ZIFF portion of the TREC/TIPSTER corpus [13]. The interface uses the Tcl/Tk X11-based toolkit [24] and the search engine uses TDB [8], implemented in Common Lisp. The use of TextFiles is not critical to the implementation; paragraphs or other segmentation units could be substituted, although this could result in units of less helpful granularity. Note that TextTiling is run in advance for the entire collection and the resulting indices stored for later use; therefore although the time for retrieval is greater than for a standard Boolean full-text query, it is not significantly so. Performance issues for indexing with passages are discussed in, for example, [22].

### RELATED WORK

As mentioned above, most information access systems have not grappled with how to display retrieval results from long texts specifically. Hypertext systems address issues related to display of contents of individual documents but are less concerned with display of contents of a large number of documents in response to a query. The Superbook system [10] shows where the hits from a query are in terms of the structure of a single, large, hierarchically structured document, but does not handle multiple documents simultaneously, nor does it show the terms of a multi-term query separately, nor does it display the frequencies graphically.

In general, document content information is difficult to display using existing graphical interface techniques because textual information does not conform to the expectations of sophisticated display paradigms, such as the techniques seen in the Information Visualizer [25]. These techniques either require the input to be structured (e.g., hierarchical, for the Cone Tree) or scalar along at least one dimension (e.g., for the Perspective Wall). The aspects of a document that satisfy these criteria (e.g., a timeline of document creation dates) do not illuminate the actual content of the documents.

Another graphical interface is that of Value Bars [4], which display relative attribute size for a set of attributes. The example in [4] shows a window listing a file directory's contents and vertical Value Bars alongside the window's scrollbar. Each horizontal slice of a Value Bar represents the size or the age of a listed file, although the attributes of the Value Bars do not align directly with window's contents nor with one another, thus precluding the perception of overlap among the displayed item's attributes. One could imagine using Value Bars for display of retrieval results by replacing the filenames with titles of retrieved documents and having the attributes correspond to the number of hits for term sets. However, the display would still not indicate term overlap or term distribution. Similar remarks apply to the Read Wear interface [17].

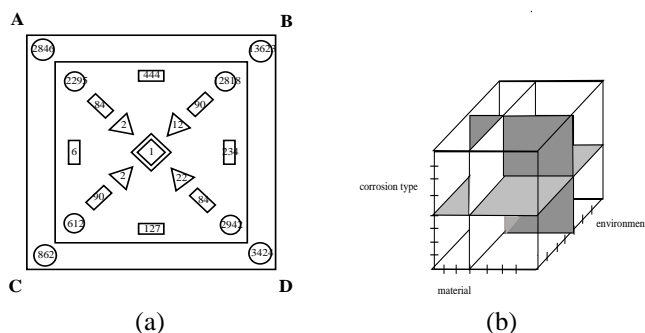


Figure 6: Sketches of (a) the InfoCrystal (b) the Cube of Contents.

Turning now to information retrieval systems, the simplest approach to displaying retrieval results is, of course, to list the titles or first lines of the retrieved documents and their ranks, and many systems do this. Existing systems that do more can be characterized as performing one of two functions: (1) displaying the retrieved documents according to their overall similarity to a query or other retrieved documents, and/or (2) displaying the retrieved documents in terms of keywords or attributes pre-selected by the user. Neither of these approaches address the issues of term distribution, frequency, and overlap that TileBars do. For reasons argued above, systems of type (1) are problematic, especially with respect to full-text collections.

Systems of type (2) show the relation of the contents of texts to user-selected attributes; these include VIBE [19], the InfoCrystal [28], the Cube of Contents [2], and the system of About *et al.* [1]. These systems require users to select the classifications around which the display is organized. The goal of VIBE [19] is to display the contents of the entire document collection in a meaningful way. The InfoCrystal [28] is a sophisticated interface which allows visualization of all possible relations among  $N$  user-specified “concepts” (or Boolean keywords). The InfoCrystal displays, in a clever extension of the Venn-diagram paradigm, the number of documents retrieved that have each possible subset of the  $N$  concepts. Figure 6(a) shows a sketch of what the InfoCrystal might display as the result of a query against four keywords or Boolean phrases, labeled A, B, C, and D. The diamond in the center indicates that one document was discovered that contains all four keywords. The triangle marked with “12” indicates that twelve documents were found containing attributes A, B, and D, and so on. The Information Crystal does not indicate information about the distribution or frequency of occurrence of the query terms within the document. Thus it is perhaps more appropriate for titles and abstracts than for full text. The Cube of Contents [2] helps the user build a query by selecting values for up to three mutually exclusive attributes (Figure 6(b)). This assumes a text pre-labeled with relevant information and an understanding of domain-dependent structural information for the document set. Again, frequency and distribution information could not

be indicated easily in this framework.

## CONCLUSIONS AND FUTURE WORK

I have introduced a new display device, called TileBars, that visualizes explicit term distribution information in a full text information access system. The representation simultaneously and compactly indicates relative document length, query term frequency, and query term distribution. The patterns in a column of TileBars can be quickly scanned and deciphered, aiding users in making fast judgments about the potential relevance of the retrieved documents. TileBars can be sorted or filtered according to their distribution patterns and term frequencies, aiding the users’ evaluation task still more. An in-depth description of an example helped show the semantic affects of various term distribution patterns. The TileBar representation should extend easily to representing media types other than text.

In the future user studies should be run to determine how users interpret the meaning of the term distributions and how they may be used in relevance feedback. It may be useful to determine in what situations the users’ expectations are not met, in hopes of identifying what additional information will help prevent misconceptions. Another kind of evaluation is currently underway [16], exploring the effects of term distribution in the TREC/TIPSTER test collection [13] on individual queries. Associated with the documents in the TIPSTER collection are a set of queries and human-assigned relevance judgments. In the past two years there has been a spate of research on passage retrieval in this collection, but the results are mixed and difficult to interpret. The main trend seems to be that some combination of scores from the full document with scores from the highest scoring passage or segment yields a small improvement over the baseline of using the full document alone. The work reported in [16] attempts to determine how term distribution and overlap affects retrieval results in this task, and in the process provides an argument for the use of a TileBar-like display. Preliminary results indicate that scores can be improved by taking individual term distribution preferences for individual queries into account.

Information access mechanisms should not be thought of as retrieval in isolation. Cutting *et al.* [9] advocate a text access paradigm that “weaves together interface, presentation and search in a mutually reinforcing fashion”; this viewpoint is adopted here as well. For example, the user might send the contents of the a TileBar window to an interface like Scatter/Gather [7] which can cluster the document subset, and display their main topics. The user could then select a subset of the clusters to be sent back to the TileBar session. This kind of integration will be attempted in future work.

## Acknowledgements

This paper has benefited from the comments of Jan Pederesen and six anonymous reviewers. I would also like to thank Robert Wilensky for supporting this line of research and Marc Teitelbaum for help in an earlier implementation.

## REFERENCES

1. M. Aboud, C. Chrisment, R. Razouk, and F. Sedes. Querying a hypertext information retrieval system by the use of classification. *Information Processing and Management*, 29(3):387–396, 1993.
2. H. C. Arents and W. F. L. Bogaerts. Concept-based retrieval of hypermedia information – from term indexing to semantic hyperindexing. *Information Processing and Management*, 29(3):373–386, 1993.
3. Jacques Bertin. *Semiology of Graphics*. The University of Wisconsin Press, Madison, WI, 1983. Translated by William J. Berg.
4. Richard Chimera. Value bars: An information visualization and navigation tool for multi-attribute listings. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 293–294, May 1992.
5. William S. Cooper, Fredric C. Gey, and Aitao Chen. Probabilistic retrieval in the TIPSTER collections: An application of staged logistic regression. In Donna Harman, editor, *Proceedings of the Second Text Retrieval Conference TREC-2*, pages 57–66. National Institute of standard and Technology Special Publication 500-215, 1994.
6. W. Bruce Croft and Howard R. Turtle. Text retrieval and inference. In Paul S. Jacobs, editor, *Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval*, pages 127–156. Lawrence Erlbaum Associates, 1992.
7. Douglass R. Cutting, David Karger, and Jan Pedersen. Constant interaction-time Scatter/Gather browsing of very large document collections. In *Proceedings of the 16th Annual International ACM/SIGIR Conference*, pages 126–135, Pittsburgh, PA, 1993.
8. Douglass R. Cutting, Jan O. Pedersen, and Per-Kristian Halvorsen. An object-oriented architecture for text retrieval. In *Conference Proceedings of RIAO'91, Intelligent Text and Image Handling, Barcelona, Spain*, pages 285–298, April 1991. Also available as Xerox PARC technical report SSL-90-83.
9. Douglass R. Cutting, Jan O. Pedersen, Per-Kristian Halvorsen, and Meg Withgott. Information theater versus information refinery. In Paul S. Jacobs, editor, *AAAI Spring Symposium on Text-based Intelligent Systems*, 1990.
10. Dennis E. Egan, Joel R. Remde, Louis M. Gomez, Thomas K. Landauer, Jennifer Eberhardt, and Carol C. Lochbaum. Formative design evaluation of superbook. *Transaction on Information Systems*, 7(1), 1989.
11. Edward A. Fox and Matthew B. Koll. Practical enhanced Boolean retrieval: Experiences with the SMART and SIRE systems. *Information Processing and Management*, 24(3), 1988.
12. Norbert Fuhr and Chris Buckley. Optimizing document indexing and search term weighting based on probabilistic models. In Donna Harman, editor, *The First Text Retrieval Conference (TREC-1)*, pages 89–100. NIST Special Publication 500-207, 1993.
13. Donna Harman. Overview of the first Text REtrieval Conference. In *Proceedings of the 16th Annual International ACM/SIGIR Conference*, pages 36–48, Pittsburgh, PA, 1993.
14. Marti A. Hearst. *Context and Structure in Automated Full-Text Information Access*. PhD thesis, University of California at Berkeley, 1994. (Computer Science Division Technical Report UCB/CSD-94/836).
15. Marti A. Hearst. Multi-paragraph segmentation of expository text. In *Proceedings of the 32nd Meeting of the Association for Computational Linguistics*, June 1994.
16. Marti A. Hearst. An investigation of term distribution effects on individual queries. Technical Report Report Number ISTL-QCA-1994-12-06, Xerox PARC, 1995. Submitted for publication.
17. William C. Hill, James D. Hollan, Dave Wroblewski, and Tim McCandless. Edit wear and read wear. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 3–9, May 1992.
18. Brewster Kahle and Art Medlar. An information system for corporate users: Wide area information servers. Technical Report TMC199, Thinking Machines Corporation, April 1991.
19. Robert R. Korfhage. To see or not to see – is that the query? In *Proceedings of the 14th Annual International ACM/SIGIR Conference*, pages 134–141, Chicago, 1991.
20. S. Kosslyn, S. Pinker, W. Simcox, and L. Parkin. *Understanding Charts and Graphs: A Project in Applied Cognitive Science*. National Institute of Education, 1983. ED 1.310/2:238687.
21. Jock Mackinlay. *Automatic Design of Graphical Presentations*. PhD thesis, Stanford University, 1986. Technical Report Stan-CS-86-1038.
22. Alistair Moffat, Ron Sacks-Davis, Ross Wilkinson, and Justin Zobel. Retrieval of partial documents. In Donna Harman, editor, *Proceedings of the Second Text Retrieval Conference TREC-2*, pages 181–190. National Institute of standard and Technology Special Publication 500-215, 1994.
23. Terry Noreault, Michael McGill, and Matthew B. Koll. A performance evaluation of similarity measures, document term weighting schemes and representations in a Boolean environment. In R. N. Oddy, S. E. Robertson, C. J. van Rijsbergen, and P. W. Williams, editors, *Information Retrieval Research*, pages 57–76. Butterworths, London, 1981.
24. John Ousterhout. An X11 toolkit based on the Tcl language. In *Proceedings of the Winter 1991 USENIX Conference*, pages 105–115, Dallas, TX, 1991.
25. George C. Robertson, Stuart K. Card, and Jock D. MacKinlay. Information visualization using 3D interactive animation. *Communications of the ACM*, 36(4):56–71, 1993.
26. Gerard Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley, Reading, MA, 1988.
27. Hikmet Senay and Eve Ignatius. Rules and principles of scientific data visualization. Technical Report GWU-IIST-90-13, Institute for Information Science and Technology, The George Washington University, 1990.
28. Anselm Spoerri. InfoCrystal: A visual tool for information retrieval & management. In *Proceedings of Information Knowledge and Management '93*, Washington, D.C., Nov 1993.
29. Edward Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Chelshire, CT, 1983.

# TextArc: Showing Word Frequency and Distribution in Text

W. Bradford Paley

Digital Image Design Incorporated (*didi.com*)

*brad@didi.com*

## Abstract

*TextArc is an alternate view of a text, tailored to expose the frequency and distribution of the words of an entire text on a single page or screen. In texts having no markup or meta-information, one of the quickest ways of getting a feeling for the content of a text is to scan through the words that are used most frequently. Knowing the distribution of those words in the text can support another level of understanding, e.g. helping to reveal chapters in a text that concentrate on a specific subject. A structure and method of displaying an entire text on a single page or screen is presented. It reveals both frequency and distribution, and provides a well-understood and organized space that works as a background for other tools.*

## 1. Introduction

TextArc was developed to help people deal with the ever-increasing influx of data they are forced to accept and integrate into their knowledge base. Much of that data comes in the form of raw text—e-mails, news stories, academic papers, and even a significant amount of data that could theoretically be categorized or indexed still comes to us as plain ASCII. TextArc was developed as a way to get an overview of a medium-sized body of raw text, e.g. the amount one might receive in a single day or week, and provide pointers into that text to let people more easily get to the things meaningful to their goals.

## 2. Existing Text Overview Methods

There are already many tools directed getting an overview of texts. Simple indices, concordances, lexicons, and other structured lists of words have been serving well for centuries. Computational linguistics techniques have recently added tools that generate automatic summaries, identify key ideas, and do semantic analysis. Several graphical techniques have also been developed to address this need, in the hopes of tapping into the vast visual processing capabilities of the human brain. Self-organizing maps have been deployed, as have multidimensional scaling techniques, to help users group similar concepts.

These approaches generally factor out one key dimension that has great meaning in a text: its original linear order. Since authors spend so much effort in crafting that order we tried to develop a technique that would

respect and build on that order. This was done in the hopes that the new view to be complimentary to existing graphical and non-graphical text overview techniques, with the conviction that the expressive variety among views available to any knowledge worker is as important as the design of any one view.

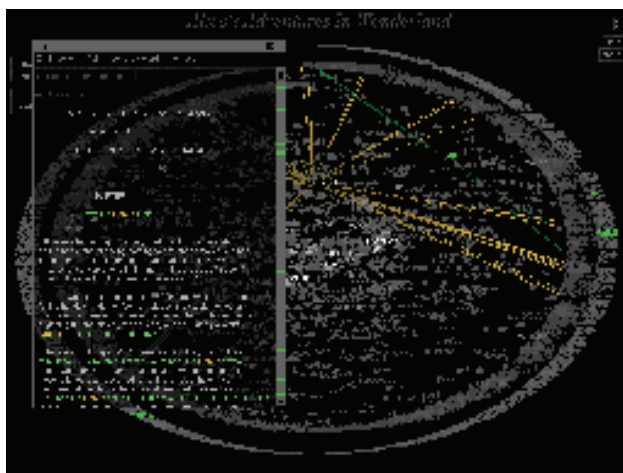


Figure 1: The interactive TextArc at [textarc.org](http://textarc.org)

## 3. TextArc Structure

A TextArc is a structure built entirely of the words in a text, generally placed in the same order that they appear in the text. Words that lose much of their meaning when taken out of context (“stop words” such as “and,” “if,” “the,” e.g.) are not initially drawn, though they may be turned back on in a control panel in the interactive version. Words are also “stemmed;” grouped together based on their word stem (e.g. “jump,” “jumped,” and “jumping” are represented by one word), though they can be ungrouped in the same control panel.

### 3.1. Text line placement

To create a TextArc first the entire text is drawn in an ellipse around the outside of the page or screen, line by line, in a tiny—potentially even unreadable—font. Lines are positioned at even increments around the ellipse: starting at the top center, keeping their baseline horizontal, and stepping each line’s starting point clockwise around the ellipse. The steps around the ellipse are scaled to make the last line appear next to the first line: the angle of each step is roughly  $360^\circ$  divided by the number of lines in the text.



# The Word Tree, an Interactive Visual Concordance

Martin Wattenberg and Fernanda B. Viégas

**Abstract**— We introduce the Word Tree, a new visualization and information-retrieval technique aimed at text documents. A word tree is a graphical version of the traditional "keyword-in-context" method, and enables rapid querying and exploration of bodies of text. In this paper we describe the design of the technique, along with some of the technical issues that arise in its implementation. In addition, we discuss the results of several months of public deployment of word trees on Many Eyes, which provides a window onto the ways in which users obtain value from the visualization.

**Index Terms**—Text visualization, document visualization, Many Eyes, case study, concordance, information retrieval, search.

## 1 INTRODUCTION

In James Joyce's *A Portrait of the Artist as a Young Man*, the word "his" appears 1,744 times. The word "her" occurs 316 times. These numbers provide little insight beyond a basic imbalance. Now consider that the most common word to follow "his" is "soul," while the most common word to follow "her" is "eyes." With this fact, the nature of the imbalance begins to emerge. Repeated elements tell us a great deal about texts—but with context more nuances and revealing themes appear.

Furthermore, the set of contextual elements often itself has a complex structure. "His soul" appears 83 times in *A Portrait of the Artist as a Young Man*, but what follows those two words? Among other phrases, "was fattening," "was festering," and "was foul"—along with "was waking," "was enriched," and "was soaring."

In this paper we introduce a new visualization technique, the word tree, that makes it easy to explore this type of repetitive context. A word tree places a tree structure onto the words that follow a particular search term, and uses that structure to arrange those words spatially. Simple interaction techniques allow the viewer to examine the ways that a particular word or phrase is used in a text, seeing broad patterns and drilling down into details.

The motivation for creating the word tree comes from our experience with user-generated visualizations on the Many Eyes site [11], which allows anyone to upload and visualize data. Since the site launched in the beginning of 2007, we have observed a growing number of attempts to visualize unstructured text. The first text visualization on the site was a tag cloud—a common technique showing word frequency. Despite the tag cloud's popularity, comments from users indicated they were sometimes as interested in usage context as raw frequency counts. The feedback prompted us to consider a technique that would retain more of the text composition for exploration.

The word tree was first made public in September 2007 and, as of March 2008, people have used the word tree to examine more than 650 texts. Because all activity on Many Eyes is public, and because many of its members write about their experiences on blogs, we have a rich record of word tree usage. We take advantage of this record to assess the word tree technique and examine the types of value that it is providing. We also describe feedback from users, which points to several promising research directions.

- 
- *Martin Wattenberg is with IBM Research, E-Mail: mwatten@us.ibm.com.*
  - *Fernanda B. Viégas is with IBM Research., E-Mail: viegasf@us.ibm.com.*

Manuscript received 31 March 2008; accepted 1 August 2008; posted online 19 October 2008; mailed on 13 October 2008.  
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

## 2 RELATED WORK

Finding ways to display and refine search results is a classic challenge in information retrieval. The problem predates computers; for centuries biblical scholars have used concordances<sup>1</sup> to see how different words occur in religious texts. The computer-age equivalent of these paper concordances is the "keyword in context" (KWIC) technique [5], in which hits are shown with the search term, or keyword, surrounded by a snippet of the text in which it occurs. Although these snippets are often arranged so that the keywords are aligned, it can be difficult to see patterns and connections in the resulting array of text.

Several visualization methods have been proposed to show word usage. Tilebars [8] provide a compressed overview of search term distribution within a document; the SeeSoft program [4] provides graphical highlighting of textual metadata. The TextArc technique [13] shows the overall distribution of every word in a piece of text. None of these, however, provides an easy way to see the different contexts in which a given word or phrase is used. Other text visualizations, such as dotplots [7] and arc diagrams [14], spotlight global patterns of repetition but do not provide a detailed view of the context of usage of particular terms.

One clever technique for displaying contextual information about a search term is the "star diagram" of Bowdidge and Griswold [2]. Designed to help developers restructure code, a star diagram shows how a particular variable is used in a program. The display uses a tree structure to display which functions are applied to that variable, which functions call those functions, and so forth up the call stack. While it is not directly applicable to plain text, the star diagram's hierarchical arrangement of context makes it a precursor to the work we describe below.

As we discuss in the next section, we propose to show context using an interactive tree structure, in which users can click on nodes to vary the level of detail. Three systems directly relate to the interaction techniques we employ. The network exploration program described in Yee et al. [15], the SpaceTree visualization [6], and Degree-of-Interest Trees (DOITrees) [9] use elegant animations to help users navigate, a design choice that we emulate. The SpaceTree and DOITrees, like our visualization, allow users to easily move up and down a hierarchy. On the other hand, our design for showing levels of detail and handling high branching factors contrasts with SpaceTree and DOITrees, and we discuss some differences in user reaction to this point.

---

<sup>1</sup> The word "concordance" is sometimes used to mean a comparison between texts. In this paper, however, we use it in the literary sense of an index that provides additional context for word usage.

### 3 DESIGN OF THE WORD TREE

A *word tree* is essentially an interactive form of the keyword-in-context (KWIC) technique. It builds on KWIC in three ways. First, it has a visual design that makes it easy to spot repetition in the contextual words that follow a phrase. Second, the design makes obvious the natural tree structure of the context. Third, it affords easy ways to explore the context further.

```
if love be rough with you , be rough with love .  
if love be blind , love cannot hit the mark .  
if love be blind , it best agrees with night .
```

Fig 1. All instances of “if love” in *Romeo and Juliet*.

A KWIC display can be thought of as a tree in disguise. Consider Figure 1, which shows the words that follow the search term “if love” in the play *Romeo and Juliet*. If one thinks of the search term as the root node, then the various distinct subsequent words define branches. In this case, because “if love” is always followed by “be,” it has just one child node, corresponding to the word “be.” The “be” node, however, has two children, one for each of the two distinct words that follow it: “rough” and “blind.” Continuing in this way one can define a tree structure that describes all the ways the search term is used.

This structure is not new. The basic idea, called a suffix tree, has for decades been an ingredient in string-processing algorithms. In fact textbook diagrams explaining suffix trees often resemble the word trees below. Despite their popularity among algorithm designers, however, suffix trees have not been used as a general visualization mechanism for search results.

#### 3.1 Many Eyes as an experimental platform

Before describing the design of the word tree, we discuss some relevant features of our experimental platform, Many Eyes. The site allows anyone on the internet to try out different visualization techniques. Previous user interviews have found a broad set of backgrounds and goals among active participants on the site [3]. Making a new visualization method available on the site is an effective way to see how it will be spontaneously used by a diverse set of people.

Experimenting in public does add some constraints, however. Unlike a supervised deployment, we cannot rely on any training for the visualization. On the contrary, if people can’t rapidly make sense of what they see, they will probably click away to a different site. Our design, therefore, put a premium on simplicity and learnability.

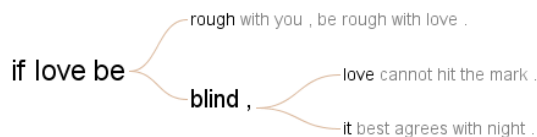


Fig 2. Word tree showing all instances of “if love” in *Romeo and Juliet*.

#### 3.2 Visual design

Figure 2 shows a simple example of a word tree, for the *Romeo and Juliet* example. The basic layout of the tree is a classic branching view. We chose this method for its instant readability. It immediately communicates to viewers that they are looking at a tree structure. Moreover, in contrast to more exotic methods such as hyperbolic trees or treemaps, it largely preserves the linear arrangement of the text.

Taking a cue from the popularity of tag clouds, we use font size to represent the number of times a word or phrase appears. The size is proportional to the square root of the frequency of the word. Using

the square root rather than a linear scale achieves two goals. First, it means the area of the word is very roughly proportional to the frequency (except for variations created by word length). Second, it leaves sufficient blank space above and below that the overall tree structure is visually obvious.

Branches of the tree continue at least until they define a unique phrase used exactly once. Instead of stopping at the first unique phrase, the tree continues until a period is reached (up to a fixed limit of tokens), so that viewers see sensible fragments of the text. To distinguish between the main tree of unique phrases, and the additional context, the former is colored black and the latter is drawn in gray.

One somewhat counterintuitive design choice is that we do not discard stopwords or even punctuation. The rationale is that prepositions and commas are often critical to understanding the meaning of a text. Leaving them out might put together phrases that mean very different things. As we discuss later, this has proved controversial with our users.

#### 3.3 Interactivity

To start exploring a word tree visualization, the user types a word or phrase into a “search” box at the top of the screen. Each time the user types the “enter” key, a punctuation mark, or a space, the tree is rebuilt. This allows a responsive feel, and for multi-word phrases lets the user quickly see if initial words in the phrase have too few hits to be worth typing additional words. Note that it would also be possible to build the tree letter by letter; however, early experiments suggested this would be distracting.

Once a word tree is shown, a user can interact with it. Moving the mouse over a particular word or phrase brings up additional information, along with a message saying that clicking will explore the tree further. Clicking on an individual word will redefine the phrase shown by the tree. This can either narrow or widen the text search. For instance, if the current phrase is “if love,” clicking on the initial “if” will re-center the tree on the phrase “if” (see Fig. 3A). On the other hand, if the user clicks on a word in a branch of the tree, such as “blind” in the branch “if love be blind,” then the tree will be re-centered on the longer phrase, “if love be blind” (Fig. 3B).

Often when looking at a tree, a user will see an unexpected or interesting word in a branch, and may want to see all uses of that word. To support this goal, a second click option is control-click, which recenters the tree on the single word clicked, no matter where it occurs in the tree. Control-clicking on “blind” in the example above will display a tree that is rooted on the word “blind” (Fig. 3C).

Someone encountering the word tree for the first time may find the result of clicking or typing unexpected. This issue is especially acute for visualizations on Many Eyes, since people can first see these visualizations embedded on totally unrelated websites, without any explicit instructions. To clarify what is changing in the tree when the user clicks or types, we built in animated transitions when possible. For example, if the user types “if love” into the search box and hits “enter,” they see the following:

1. The tree is built when the space after “if” is typed, fading gently into place to avoid an abrupt “flash” of information.
2. After “enter” is typed, an animation occurs, in which the branch for “love” becomes bigger and the other branches fade away.

Additional adjustments occur automatically. The text scale changes to put as many words as possible on the screen, while making sure that the largest words are readable. For repetitive texts, the word tree can sometimes take a large amount of horizontal space, so scrollbars are provided. Overall, the fluid feel of the interaction is similar to that of the radial tree explorer discussed in Yee *et al.* [15] and the SpaceTree system [6].

One option is to show context words trailing the given search phrase, another option is to show words that precede that phrase. Switching between these options is not animated, since there is not a sensible way to interpolate between them. Another free parameter in the visualization is the order of the branches beneath each node. The

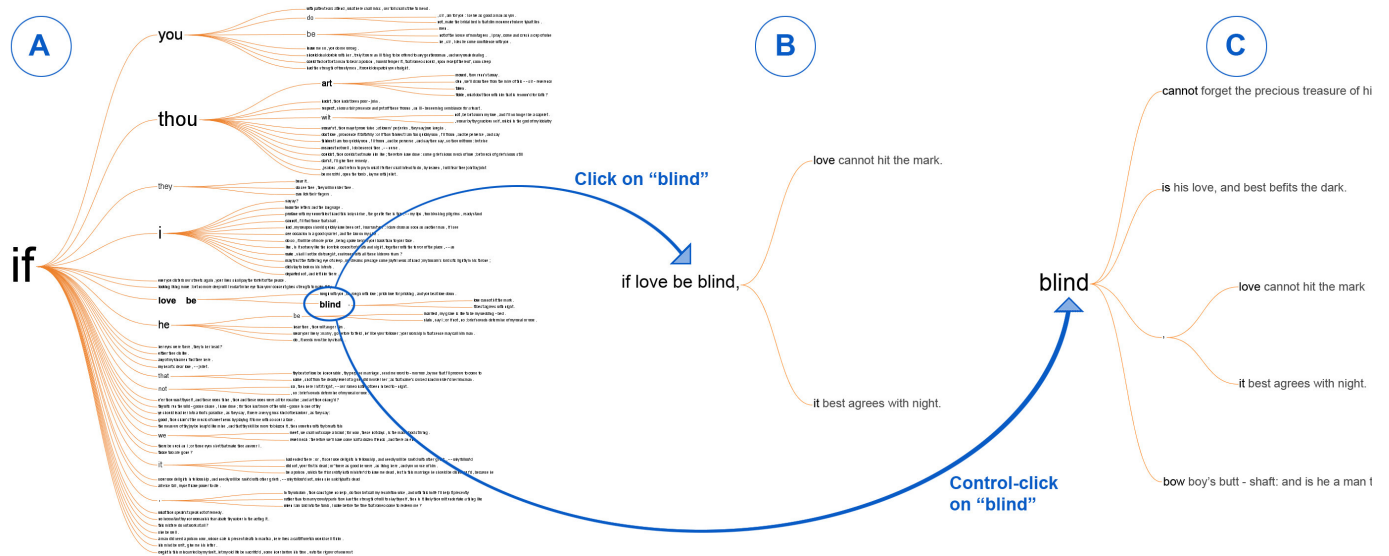


Fig 3. Sequence showing some of the interaction options in the word tree. In figure A, the user has typed the word “if” in *Romeo and Juliet*. In B, the user has clicked on “blind,” which appears in one of the branches under “if.” This causes the visualization to recenter to the longer phrase “if love be blind.” In C, the user Control-clicks on “blind,” which causes the visualization to recenter to blind by itself, revealing that there are additional phrases after this term.

Many Eyes word tree provides a choice among three options. The branches can be arranged alphabetically (making it easy to scan for particular words), by frequency (so the largest branches are first), or by order of first occurrence in the text (the default option, since it often produces a tree that best reflects the underlying text.) As with clicking, when the user switches between two of these options the word tree animates smoothly to help make clear what is changing.

As the user interacts with the tree—she may click on a branch, recenter the tree, choose a different search term, etc.—the word tree tracks of the sequence of actions just as a web browser does. This allows the user to click on browser-like “back” and “forward” buttons to review her previous steps in the visualization. This feature helps users quickly switch between desired states for comparisons and easily retreat from navigational dead ends.

As with all visualizations on Many Eyes, users can set particular states and make comments. In doing so, they may wish to point to particular items on the visualizations. To support this, users can set the visualization to a “highlighter mode,” where clicking on words will not cause a recentering of the tree, but instead highlight words with translucent brown circles. Thus a user can leave a comment like, “Note the position of God in this context,” and highlight “God” so that other readers do not need to search for where it occurs.

Finally, the word tree does not provide any sort of “overview” of the text nor does it present an initial search term for viewers to start from. In this way, the visualization resembles an information retrieval interface, driven by a search term rather than starting with an overview. The reason for this design choice is that without a search term, there is no obvious entry point—several alternatives with suffix-tree-like beginnings were attempted, but seemed busy and uninformative. A future version might try to automatically find a good starting point: perhaps a tree centered on the most frequent terms, a tree that shows the highest number of separate branches, or a tree with the deepest branches. Having a default start point might solve certain problems. For instance in the current system, unless the creator of the word tree actively sets an initial search term, the visualization will look blank to subsequent viewers on the site. Another limitation of not having an overview is that users need to know a bit about the underlying data to make sure that they look for words that appear in the text. Many other interactive features are

possible. We discuss these in the sections on user feedback and future work.

#### 4 IMPLEMENTATION CONSIDERATIONS

The current implementation of the Word Tree on Many Eyes is a Java applet, written using JDK 1.4. It is engineered to handle texts with up to 1,000,000 tokens. (In addition to being a pleasingly round figure, this is the approximate number of tokens in the King James Bible, probably one of the most-visualized text on Many Eyes.) In this section we discuss some of the implementation details and decisions that allow the applet to scale—both visually and in performance—to a million tokens.

The data structure behind the word tree—that is, the hierarchical structure of the context words—is well-known to computer scientists as a “suffix tree.” In our context the practical bound on performance is memory rather than CPU cycles: constructing the tree is fast (at least for a million-token text) as long as there is sufficient memory. Java applets often have limited heap space, as low as 64MB. Although this may seem more than adequate for holding a million-node tree, it is actually a serious constraint due to the memory-intensive nature of Java objects. To get around the problem, we do not create a suffix tree for the entire text, but rather create the suffix tree on the fly, a new one for each phrase typed in. In practice this saves a significant amount of memory; for instance, in the King James Bible (about 1,000,000 tokens), the word tree for “the” has only about 64,000 leaves. This complicates effects such as animated transitions, but permits the feeling of instant feedback we desire.

In addition to the data-level scaling, two issues arise in scaling the tree visually. The first is that the total number of branches is huge compared to the screen size. When there are tens of thousands of leaves to a tree, there is no sensible way of displaying all of these on a screen that is a few hundred pixels high. We resolve this issue by a standard “level of detail” method. As the geometry of the tree is defined, when it is determined that a subtree takes up less than 3 pixels of vertical space, we do not draw the entire subtree. Instead, we find the deepest branch, and draw that. By doing so, we show the overall shape of the tree, but do not draw more than necessary. This simplifies the display and also keeps the number of rendered objects low enough that smooth animated transitions are possible.



One might contrast this with the method used by Bederson et al [6] in the SpaceTree. In that visualization, once items become too small to see, subtrees are replaced by icons or simplified views that indicate the overall breadth and depth. Note that our method does not need to communicate a general sense of breadth, since we only apply level of detail calculations when the breadth of a subtree is negligible compared to the breadth of the overall tree.

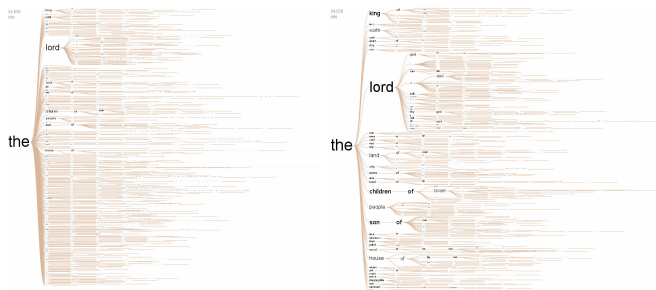


Fig 4. Two versions of the word tree: on the left, all branches under “the” are displayed, causing most words to become unreadable. On the right, we use a “level of detail” method, showing a subset of branches.

A second design issue arises for common words in a large text: for instance, in the word tree for “the” in the King James Bible, which occurs 64,028 times. 3,604 distinct words follow “the” in this text, with the most common being “Lord” at 11% of the total. That is enough to be visible, but the second most common word, “son,” takes up only 2.3%, barely enough to be legible. As a result, if all words were shown—or even if only a few were shown with sizes proportional to their frequency—then almost none would be readable.

To handle this problem, we compromise and show only the largest branches, with the number defined so that only branches with at least 1% of the total leaves are included. In the case of the King James Bible and “the,” this means that words like “son,” “children,” and “king” are legible. This method of pruning is continued recursively for sub-branches. Note that the result is a genuine compromise, since a user can’t immediately deduce the true proportional usage of a word in context, since we have removed the “long tail” of infrequently seen words. It is worth emphasizing the difference from the kind of data used in SpaceTree, where the prototypical use case was an organizational chart with branching factors in the dozens, not thousands. An approach similar to our is taken by the DOITree described in [9], which does collapse nodes of lesser significance to make room for others; on the other hand—but as with the SpaceTree, the DOITree strives for readability of individual items over a sense of overall breadth,

## 5 SPONTANEOUS USAGE ON MANY EYES

As of this writing, users have created 658 word trees on Many Eyes. This section describes some of these word trees and the types of data they have been used for<sup>2</sup>. Because of the ease of access to Many Eyes, some of these word trees represent undirected “tire-kicking” and aimless experimentation. In other cases, however, users had specific goals in mind and described their actions in detail. We found these more detailed cases through two avenues: on and off site. User comments and actions on the site led us to several examples. We also performed searches on Google, which reported

<sup>2</sup> In this section we reprint some visualizations created by registered users of Many Eyes. As part of registration, all users gave permission for us to create such copies.

more than 300 references “word tree” with either “manyeyes” or “many eyes” outside of the ibm.com domain.

The search led us to some of the most detailed reports. In one, entitled “Using Word Tree Visualization for Checking Title Consistency” [1], a blogger wrote a 1,007-word essay describing his use of the visualization. His initial task was creating a series of title-like summaries for stories in the Bible. He decided to use the word tree to visualize his collection, and even went to the trouble of adding special “+Start+” and “+end+” tokens to his titles so that they would not run together in the tree. After doing this, he reported:

*Looking at the frequency-sorted suffixes for “+start+ Jesus warns”, i see a large group under “against”, and a number under “about”, but also a single instance, “Jesus warns of coming judgment”. Because the third word is “of” rather than “about”, it stands apart from the other instances which really share the same concept.*

He went on to describe how he could rewrite the title to become more consistent. Along with describing the value he derived from the visualization, he also provided a very detailed description of the word tree and how he interacted with it.

*... clicking on “teaches” narrows the view further (which you pretty much have to do to see the details: results over 30 or 40 aren’t really visible). One advantage of this representation is that it gives you some help in knowing what to explore (in user interface terminology, an affordance). Though i can’t see all the details without zooming in, i can see a significant cluster of titles starting with “Jesus warns”, and if that’s interesting, i can click on “warns” to zoom in and see those 18 titles.*

This description of the effect of the too-small-to-read type is very interesting, because it stands in contrast to a result reported for the SpaceTree. In [6], Grosjean et al. emphasize that their users “rejected bluntly” unreadable type, and the authors of the paper created a new type of icon as an alternative to scaled-down text. It is unclear whether this blogger would have preferred the SpaceTree icons; it is also possible that there is great variability among individual users’ preferences, or that the different branching factors in the SpaceTree data versus our suffix trees are amenable to different designs.

Finally, this blogger also made a suggestion for a future feature, namely drilling down for more details:

*What would be really great would be to turn this from a visualization into a navigation system, so once i’ve drilled down to “Jesus warns against ...”, then i could select a title and actually view the pericope text.*

While this essay-length blog entry was unusual, the creativity and energy behind it were echoed in the actions of many other users. We now describe some of the general trends in how they used the word tree.

### 5.1 Visualizing the spoken word

In April of 2007, former U.S. Attorney General Alberto Gonzales testified before the Senate Judiciary Committee on his role in the dismissal of several U.S. attorneys. As the transcript of Mr. Gonzales’s testimony became available on the web, one of the authors in this paper visualized the text as a word tree on Many Eyes. The visualization clearly showed a number of times when Mr. Gonzales had used expressions such as “I don’t recall,” “I don’t know,” “I don’t think” while testifying (Fig. 5).

In September, the Gonzales visualization was featured on the front page of Many Eyes—a prominent spot on the site. Within 90 minutes of it being featured, another user had created a new word tree, entitled “William ‘I don’t recall’ Jefferson Clinton Testimony in Sexual Harrassment Lawsuit that led to his impeachment” [sic]. Here



Fig 5: Alberto Gonzales' testimony in 2007.

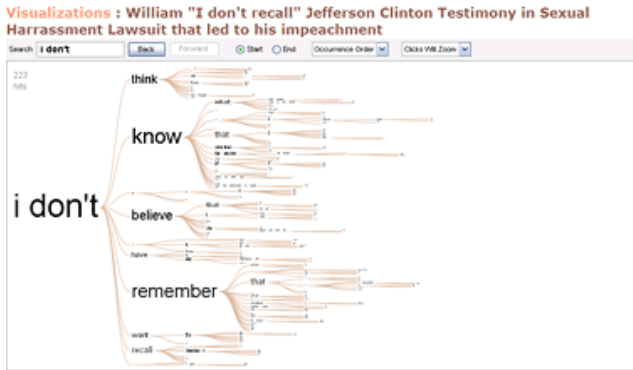


Fig 6: Bill Clinton's testimony in 1998.

too, the user set up the visualization so that the default view would show the number of times Mr. Clinton had said “I don’t know,” “I don’t remember,” “I don’t think,” etc (Fig 6). In short, both visualizations gave a clear portrait of evasive testimony.

Given that these scandals focused on politicians at opposite ends of the political spectrum, the visualizations take on an evident spin, with even the act of their creation suggesting political affiliations and beliefs. This sort of contribution to “counter” someone else’s creation on Many Eyes indicates that users are integrating these tools in their communicative practices. Far from being dispassionate representations of data, the two “I don’t recall” word trees are part of a political conversation, a dialog happening through visualization.

The ability to visualize political transcripts has resonated with our user base. Since the word tree was launched during the preparation for the 2008 U.S. presidential election, users frequently created word trees of political speeches, debates among candidates, and media coverage of the election.

Emotionally charged transcripts such as congressional hearings and political speeches are not the only kind of transcripts being visualized on Many Eyes. Even communities that are traditionally immersed in numerical data, such as financial analysts and investors, have started to explore the possibilities of using word trees to visualize transcripts. Earlier this year Seeking Alpha, a well-known online column of stock market opinion and analysis, embedded both a word tree and a tag cloud of the transcript of an earnings conference call on its site and invited readers comment on their value.

The post quickly generated a number of comments, not all of them in approval of the experiment. Some felt that the word tree was more helpful than the tag cloud because it kept the structure of the text, while others mentioned that it was easier not to use visualization at all:

*Just give us the text, we know how to find (Ctrl+f)*

As with the Blogos author, a common request was for the ability to click on an item in the visualization and see the places in the raw transcript where that item appears.

## 5.2 Visualizing the written word

The word tree was designed to handle texts of up to a million tokens, and to demonstrate this we created a visualization of the King James Bible, which contains 1,007,116 words and punctuation marks. Once the visualization was posted on the site, it was quickly picked up by a group of users interested in religious texts. The reaction was positive; this comment, unusual for visualizations in general, typified the response:

*This is a new tool to teach the Bible's truth. God bless you.*

Other users promptly explored various entryways into the text, looking for expressions such as “days of thy,” “my love,” and “love the lord” (Figure 9). As previously noted [13], visualizations of religious data have been a regular occurrence in Many Eyes since the site was launched. Perhaps it is not surprising that this community would be excited to experiment with the analytical possibilities of the word tree.

Users have also created numerous word trees of literary works, musical lyrics, and academic papers. An interesting trend is the visualization of online social activity. Some users have started visualizing collections of Twitter posts, blog posts, and newsgroup discussions. It seems that, like tag clouds, word trees might be helpful in giving people a quick sense of distributed activity online.

## 5.3 Visualizing structure

Although the word tree was designed to analyze unstructured text, it is based on a visualization of abstract tree structures. Users quickly caught on to the possibility of visualizing structured data and started specially formatting data in ways that would induce the word tree to show tree-structured information.

One person uploaded a data set of Greek nominal suffixes used in the New Testament with full nominal morphology. Because this data set is not a regular text passage but rather a list of words spaced out into individual letters, the word tree looks cryptic (see Fig. 7). If, for example, a user does a search for, NPM (nominative, plural, masculine words), they will see the suffix tree is dominated by –OI and –ONTES. This arrangement shows that the large majority of nominative, plural, masculine words in Greek end in –OI or –ONTES.

Another user created a data set to show the different pathways to the U.S. Presidency. The data set lists the names of 19 American presidents and the sequence of titles held by each one of them (Fig.

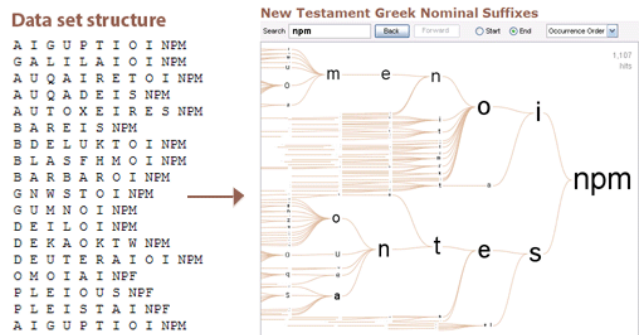


Fig 7. Data set and word tree of Greek nominal suffixes in the Bible. Here, “npm” refers to nominative, plural, masculine nouns.

## Data set Structure

President Governor George\_W\_Bush  
President Governor Governor Attorney\_General Clinton  
President Vice\_President Ambassador CIA Liason Representative George\_HW\_Bush  
President Governor Reagan  
President Governor State\_Senator Carter  
President Vice\_President Representative Ford  
President Vice\_President Senator Representative Nixon  
President Vice\_President Senator Representative Johnson  
President Senator Representative Kennedy  
President General Eisenhower  
President Vice\_President Senator Truman  
President Governor Secretary\_of\_the\_Navy State\_Senator Franklin\_Roosevelt  
President Secretary\_of\_Commerce Humanitarian Hoover  
President Vice\_President Governor Coolidge  
President Senator Lt\_Governor State\_Senator Harding  
President Governor University\_President Wilson  
President Secretary\_of\_War Governor\_General\_of\_Phillippines Federal\_Judge

## Pathways to the Presidency

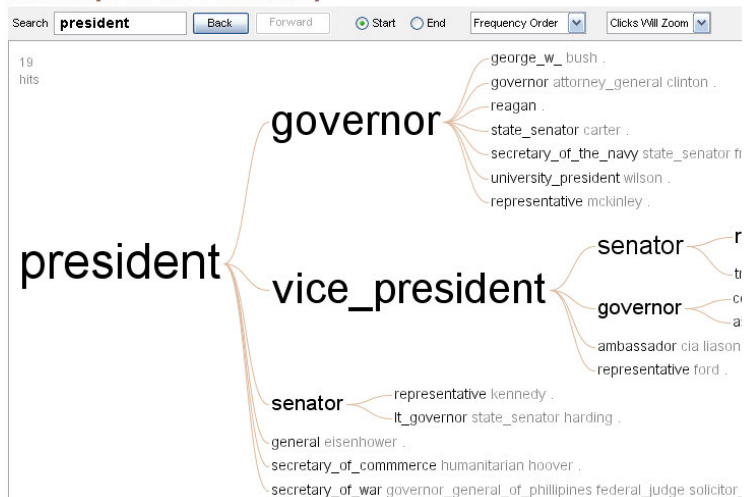


Fig 8. Data set and word tree of pathways to the U.S. presidency.

8). The word tree reveals that, whereas there were ten presidents who had served as governors, only five had previously been senators—Harding was the first and Kennedy the last. One might wonder what this historical fact spells for the 2008 elections.

Among other things, these examples show that at least some users can understand how the data structure maps to the visualization technique well enough to reverse-engineer the visualization to serve new purposes.

### 5.4 User feedback

Working on a public web site for visualization gives us the ability to collect user feedback on the new techniques we launch. The word tree has generated many comments about what users found helpful, problematic, and even missing. The response has generally been positive with a number of suggestions for improvements.

Users were excited about the ability to visualize large pieces of text while keeping some of the context around keywords. Several people remarked on how much they liked the capability of “zooming” in and out of specific branches of the tree. When comparing the word tree to the tag cloud, users felt that the word tree allowed them to engage in deeper analysis of the text.

On the other hand, not all feedback was positive and users suggested many new features. Several of these are relatively minor changes. The most commonly requested feature was for an option to ignore punctuation, and sometimes stopwords as well. Because sometimes these tokens take a lot of room in the visualization, users would prefer to have the option of turning stop-words and punctuation on and off. In addition, some users also wanted more context, in particular the ability to place their search terms “in the middle” of the word tree, surrounded by both the preceding and following context a given phrase.

The second main group of suggestions centered on the ability to drill down from the word tree into a plain view of the text. A related request is to see the locations of all the uses of a particular word or phrase, perhaps by drawing lines from the nodes of the word tree into a vertical line representing the extent of the text. Finally, several users requested the ability to filter the text, showing word trees of just particular sections.

## CONCLUSION AND FUTURE WORK

The Word Tree is sufficiently flexible and engaging that hundreds of people have used it to examine data on the Many Eyes site. The technique presented here is extremely simple, however, and there are many natural extensions. We note some promising future directions in this section, and then conclude with a brief discussion of what the popularity of the Word Tree may tell us about visualization of text on the web.

As described above, users have provided a long list of desired features. Many of these, such as drilldown from the tree into the original text, are straightforward to implement, but a few point to larger research directions. One kind of implicit request from our users regards comparisons. Many users like to look at different texts, or different phrases in the same text. For example, people will create word trees of different sections of the Bible, or might look at the context for contrasting words like “his” or “her” and what follows them. To support this type of comparison, it would be nice to overlay two trees on top of one another, perhaps with some sort of color coding.

A second idea, suggested by several users including [10] is to show a “net” of the words that connect two phrases. In other words, a user could type in “Romeo” and “Juliet” when studying Shakespeare, and see all chains of words (of less than a given length) that connect the two. These chains would not form a tree, of course, but a sort of network anchored at the two ends. Exactly how best to display and interact with such a net is an interesting problem.

Another natural direction is to handle much larger data sets. Doing so would probably require an offline calculation of a suffix tree, which would be stored on a server. Algorithms for doing such computations exist—they are handy in biology, for example—but may have to be modified to handle level-of-detail issues.

One problem identified users is that there is no natural entry point into the visualization: the viewer starts by seeing a blank screen. It might be worthwhile combining a word tree with other text visualizations, whether a tag cloud or a more complex system such as Jigsaw [12] that could provide an initial analysis of entities of interest.

Finally, the hundreds of word trees created on the Many Eyes site point to a broader implication: people are hungry for new ways to look at unstructured data. Predicting the exact use cases for these visualizations is difficult. Before seeing the first bible visualizations

on Many Eyes, for instance, we would not have guessed at the popularity of religious analyses. Given the broad demand for text visualizations, however, it seems like a fruitful area of study.

### ACKNOWLEDGEMENTS

The authors thank Frank van Ham, Jesse Kriss, Matt McKeon, Lee Byron, and Eric Gilbert for helpful suggestions. In addition, we are grateful to the users of Many Eyes for their creativity and willingness to provide feedback on an experimental visualization technique.



Fig 9. Word tree of the King James Bible showing all occurrences of “love the.”

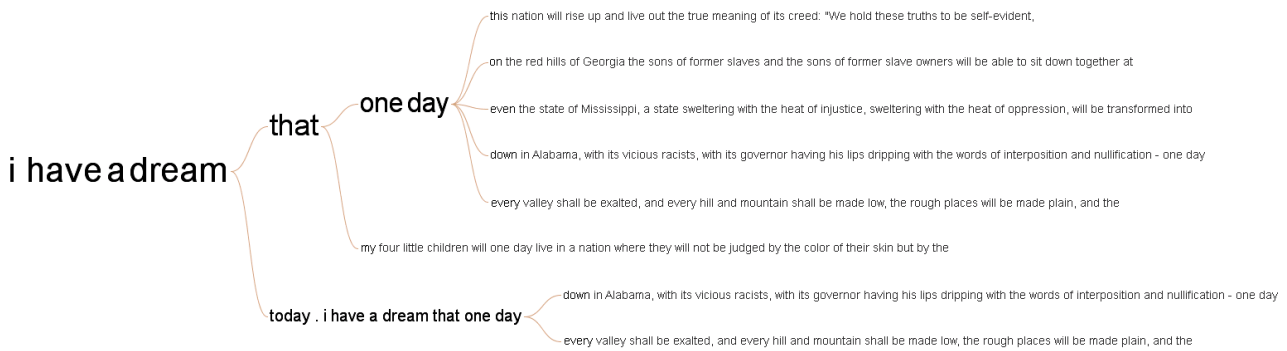


Fig 10: Word Tree showing all occurrences of “I have a dream” in Martin Luther King’s historical speech.

## REFERENCES

- [1] Boisen, Sean, "Visualizing Bible Data at Many Eyes." Blog entry. <http://semanticbible.com/blogos/2007/01/25/visualizing-bible-data-at-many-eyes/>
- [2] Bowdidge, R. and Griswold, W. (1998) Supporting the restructuring of data abstractions through manipulation of a program visualization. *ACM Transactions on Software Engineering and Methodology*. 7(2) 109-157.
- [3] Your Place or Mine? *Visualization as a Community Component*. Catalina M. Danis, Fernanda B. Viégas, Martin Wattenberg, Jesse Kriss. CHI, 2008.
- [4] Eick, S., Steffen, J., and Sumner, E. (1992) Seesoft-A Tool for Visualizing Line Oriented Software Statistics. *IEEE Transactions on Software Engineering*, 18(11), pp. 957-968.
- [5] Fischer, M. (1966). The KWIC index concept: A retrospective view. *American Documentation*. 17 (2) pp. 57 -70
- [6] Grosjean, J., Plaisant, C., Bederson, B. (2002). SpaceTree: Supporting Exploration in Large Node Link Trees, Design Evolution and Empirical Evaluation. *IEEE Symposium on Information Visualization*.
- [7] Helfman, J. (1996) Dotplot patterns: a literal look at pattern languages. *Theory and Practice of Object Systems*, 2(1) pp. 31 – 41.
- [8] Hearst, M. (1995) TileBars: Visualization of Term Distribution Information in Full Text Information Access, *ACM Conference on Human Factors in Computing Systems*.
- [9] Heer, J. and Card, S. (2004) DOTrees Revisited: Scalable, Space-Constrained Visualization of Hierarchical Data. *Proc. Advanced Visual Interfaces*
- [10] Hurst, M. "Word Trees." Blog entry. [http://datamining.typepad.com/data\\_mining/2007/09/word-trees.html](http://datamining.typepad.com/data_mining/2007/09/word-trees.html)
- [11] Paley, B. (2002) TextArc. <http://www.textarc.org>
- [12] Stasko, J., Gorg C., and Liu, Z. "Jigsaw: Supporting Investigative Analysis through Interactive Visualization", *Information Visualization*, Vol. 7, No. 2, Summer 2008, pp. 118-132
- [13] Viégas, F.B., Wattenberg, M., van Ham, F., Kriss, J., & McKeon, M. (2007) Many Eyes: A Site for Visualization at Internet Scale. *IEEE Symposium on Information Visualization*.
- [14] Wattenberg, M. (2002) Arc Diagrams: Visualizing Structure in Strings. *IEEE Symposium on Information Visualization*.
- [15] Yee, K, Fisher, D., Dhamia, R., and Hearst, M. (2001) Animated Exploration of Graphs with Radial Layout. *IEEE Symposium on Information Visualization*

# Arc Diagrams: Visualizing Structure in Strings

Martin Wattenberg  
IBM Research  
One Rogers Street  
Cambridge MA 02142  
mwatten@us.ibm.com

## Abstract

*This paper introduces a new visualization method, the arc diagram, which is capable of representing complex patterns of repetition in string data. Arc diagrams improve over previous methods such as dotplots because they scale efficiently for strings that contain many instances of the same subsequence. This paper describes design and implementation issues related to arc diagrams and shows how they may be applied to visualize such diverse data as music, text, and compiled code.*

**Keywords:** string, sequence, visualization, arc diagram, music, text, code

## 1. Introduction

From text to DNA to melodies, many data sets come in the form of a string, or sequence of symbols. Just as with quantitative data, it is often desirable to perform graphical exploratory analysis on a string to find important structural features.

One way to reveal a string's structure is to exploit the fact that sequences often contain significant repeated subsequences. Melodies, for instance, are usually based on combinations of smaller repeated musical passages; text has repeated words and phrases. A natural way to visualize structure is to use these repeated units as signposts.

Several existing methods use repetition to visualize string structure, but each has significant drawbacks for complex strings. In this paper we introduce the *arc diagram*, a new visualization method for representing sequence structure by highlighting repeated subsequences. We describe how arc diagrams can find patterns in text, compiled code and, most fruitfully, in musical compositions.

## 2. Existing methods for string visualization

Many methods have been proposed to display string structure visually. The H-Curve and W-Curve [HR83, W93] transform sequences into curves in 3D space. Although such curves are capable of showing fine detail,

they can be hard to interpret and it can be difficult to spot smaller repeated substrings. The "Chaos Game" representation of a sequence [J90], in effect a 2D histogram depicting the frequencies of various motifs, is efficient for showing which small substrings are frequently repeated, but can run into difficulties distinguishing long subsequences with similar beginnings. Moreover, chaos game representations remove much ordering information, making them unsuitable for domains where ordering matters.

Another method, popular in analyzing DNA sequences, is the dotplot [CH92]. A dotplot is a visual auto-correlation matrix; in its simplest form, a string of  $n$  symbols  $a_1a_2\dots a_n$  is represented by an  $n \times n$  image in which the pixel at coordinates  $(i,j)$  is colored black if  $a_i=a_j$  and white otherwise. This image often provides a good picture of the string's structure, with repeated subsequences showing up clearly as diagonal lines. In many respects dotplots are an excellent visualization method: They can handle very large data sets, are resistant to noise, and can show both small and large-scale structures. However, the matrix-style presentation of a dotplot means that if a substring is repeated  $n$  times, it will give rise to  $n^2$  corresponding visual features. As a result, dotplots can be confusing when applied to strings with frequently repeated substrings.

One non-visual method of describing a long string is to summarize it by describing which subsequences are repeated. For instance, musicians have long described the global structure of musical compositions by summaries such as "AABB" (meaning a subsequence, denoted by A, is repeated and then followed by a different subsequence, B, that is also repeated.) This simple symbolic notation is easy to understand and provides a broad overview of the data, but obliterates smaller details.

It is natural to seek a visual analogue of this notation. Music theorists, starting with Heinrich Schenker, have used a system of hand-drawn arcs to indicate structural units (see, for example, [S69]). However Schenkerian diagrams, which are intrinsically subjective and manual, are unsuitable for automation or for showing features on multiple scales. One commercial software package, TimeSketch [T02], uses half-disks to delineate different

sections of a piece, coloring related passages with the same color to indicate musical form. The TimeSketch software requires human definition of related passages, and because it uses color for differentiation does not scale well for sequences that have many different related passages.

### 3a. The arc diagram

This paper introduces the *arc diagram*. An arc diagram generalizes the musical AABB notation by using a pattern-matching algorithm to find repeated substrings, and then representing them visually as translucent arcs. Unlike a TimeSketch diagram, an arc diagram can be constructed automatically and can represent the structure of a sequence with many different repeated subsequences and multiple scales of repetition. Unlike a dotplot, it can efficiently represent sequences where individual subsequences are repeated many times.

An arc diagram is built around the idea of visualizing only a subset of all possible pairs of matching substrings. By choosing to highlight just the subsequences essential to understanding the string’s structure, the method can convey all critical structure while avoiding the quadratic scaling problem of a dotplot. We now define these “essential” substring pairs for a given string  $S$ .

**Definition 1.** A *maximal matching pair* is a pair of substrings of  $S$ ,  $X$  and  $Y$ , which are:

1. *Identical.*  $X$  and  $Y$  consist of the same sequence of symbols.
2. *Non-overlapping.*  $X$  and  $Y$  do not intersect.
3. *Consecutive.*  $X$  occurs before  $Y$ , and there is no substring  $Z$ , identical to  $X$  and  $Y$ , whose beginning falls between the beginning of  $X$  and the beginning of  $Y$ .
4. *Maximal.* There do not exist longer identical non-overlapping subsequences  $X'$  and  $Y'$  with  $X'$  containing  $X$  and  $Y'$  containing  $Y$ .

For example, in the sequence “123a123”, the two “123” substrings form a maximal matching pair, but the two “12” substrings do not.

It is tempting to base a visualization method on maximal matching pairs alone, but an awkward situation arises when a pattern is repeated many times in immediate succession. For instance in the string 10101010, the only maximal matching pair consists of the first and last “1010” substrings, implying that the string has two main structural components. In a sense, this division into two large substrings is spurious; it would be more accurate to describe the string as composed of four small repeated

units. This is the motivation for the following two definitions.

**Definition 2.** A *repetition region*  $R$  is a substring  $R$  of  $S$  such that  $R$  is made up of a string  $P$  repeated two or more times in immediate succession. Each repetition of  $P$  is called a *fundamental substring* for  $R$ .

For example, in the string ABC010101, the substring “010101” is a repetition region. Each of the “01” substrings is a fundamental substring.

The next definition specifies the precise set of substrings that will be used to construct an arc diagram.

**Definition 3.** An *essential matching pair* is a pair of substrings of  $S$ ,  $X$  and  $Y$ , which are:

1. A maximal matching pair not contained in any repetition region,
2. *Or*, a maximal matching pair contained in the same fundamental substring of any repetition region that contains it,
3. *Or*, two consecutive fundamental substrings for a repetition region.

We are now ready to define the arc diagram for a string  $S$  of length  $N$ . First, define a mapping from the string to the  $x$ -axis, with the position of the  $m$ th symbol at the point  $(m/N, 0)$ . Under this mapping, a substring  $T$  of  $S$  corresponds to an interval on the  $x$ -axis. Now, for each essential matching pair  $(X, Y)$  in the string, connect the corresponding intervals on the line with a thick semi-circular arc (Figure 1). Precisely, the interior semi-circle connects the end of the interval for  $X$  with the beginning of the interval for  $Y$ , and the exterior semi-circle connects the beginning of the interval for  $X$  with the end of the interval for  $Y$ . The height of the resulting arc is thus proportional to the distance between the two substrings.



The arc makes it obvious where the repeated subsequences are. (By comparison, imagine finding this repetition without the arc as a cue; it would be laborious.)

Because of the “consecutive” condition of Definition 1, if a particular subsequence is repeated more than once, the diagram connects only consecutive repetitions with an arc. (See Figure 2.) The fact that only consecutive pairs are connected rather than every possible pair is what allows arc diagrams to scale more efficiently than dotplots: when a subsequence is repeated  $n$  times, an arc diagram will contain  $n-1$  arcs, while a dotplot would display  $n^2$  diagonal lines. (See section 3b for an example.)



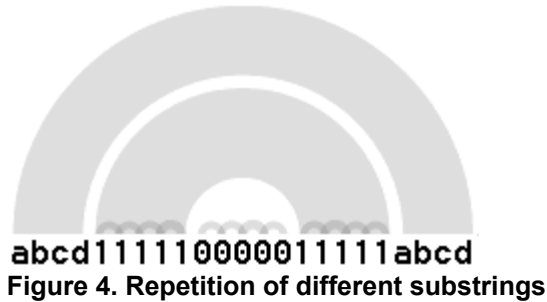
**Figure 2. A substring repeated three times**

As a simple example of visualizing strings with repetition regions, take the sequence 10101010101010. Here the only essential matching pairs are those that satisfy part 3 of Definition 3—that is, they are the fundamental substrings of the form “10”. The diagram for this sequence would look like the picture in Figure 3.



**Figure 3. Immediate repetition**

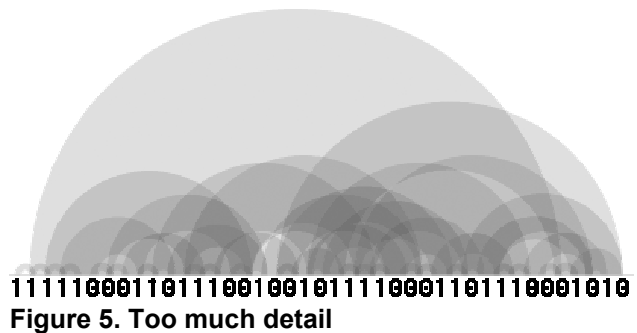
Most sequences will contain several different repeated subsequences, including overlapping sequences at multiple scales. To make a diagram for such a sequence, we overlay all the appropriate arcs with a degree of translucency so no match is completely obscured. For instance the sequence `abcd11110000011111abcd` produces the diagram in Figure 4:



**Figure 4. Repetition of different substrings**

In Figure 4 we begin to see how a pattern of matches can provide a bird's-eye view of the sequence's structure. It is visually obvious that at the macro level the sequence is symmetric. The translucency further reveals a highly repetitive substructure, without interfering with macro-level interpretation. This technique is similar to that used in Jerding and Stasko's Information Mural [JS95].

A long sequence made from a small set of symbols will always contain many small repeated sequences, which may be of no significance. Worse, the arcs connecting these small sequences may obscure significant large-scale repetitions. Figure 5 shows an example where too many small repeated subsequences cause an uninformative jumble.



**Figure 5. Too much detail**

One way of reducing this complexity is to filter by subsequence length, displaying only repeated subsequences that are longer than a given limit. For instance, Figure 6 shows the same sequence but this time set to filter out repeated subsequences of fewer than 10 symbols. The result is a simple diagram that highlights a single repeated region of 15 symbols—the kind of large-scale repetition that is unlikely to occur by chance.





Figure 6. Displaying only large-scale repetition

### 3b. Comparison with a dotplot

As mentioned above, the reason we do not connect every possible matching subsequence is so that the resulting diagram scales efficiently from a visual perspective. To see how this works, consider two visualizations of the same string, a dotplot (Figure 7) and an arc diagram (Figure 8). The string visualized contains many repetitions of two substrings, which results in considerable visual clutter in the dotplot. Two other substrings are each repeated once, a feature that is difficult to spot in the dotplot. The arc diagram, however, shows all the repeated structures clearly.

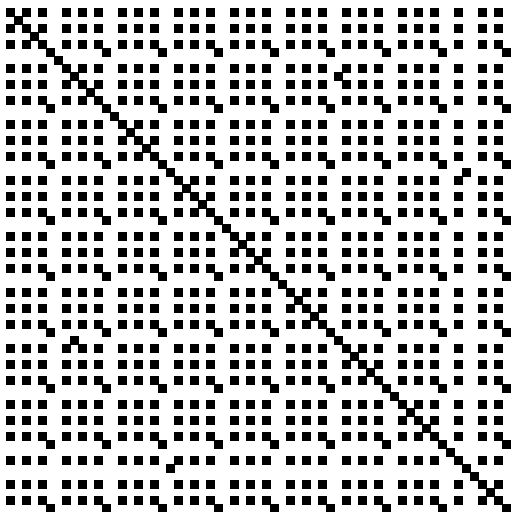


Figure 7. Dotplot of a synthetic sequence



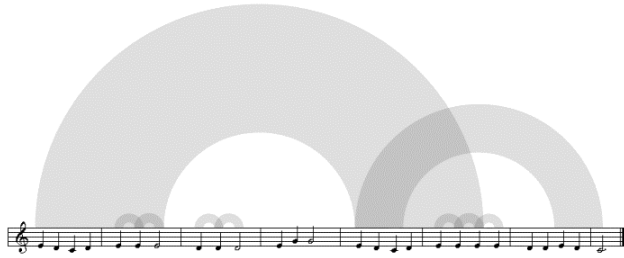
Figure 8. Arc diagram, same sequence as Fig. 8

### 3c. Implementation

The program used to create the arc diagrams in this paper is written in Java, runs efficiently on a low-end (266 Mhz Pentium II) machine, and can create diagrams of sequences of several thousand symbols within seconds. To enumerate repeated patterns, a suffix tree is constructed and traversed twice. In the first pass, repetition regions are identified and in the second pass potential matching substring pairs are tested to see whether they are essential according to the criteria of Definition 3. The arc diagram code has also been used in an online applet, "The Shape of Song" [W01], that allows users to create arc diagrams for any MIDI format music file available on the web.

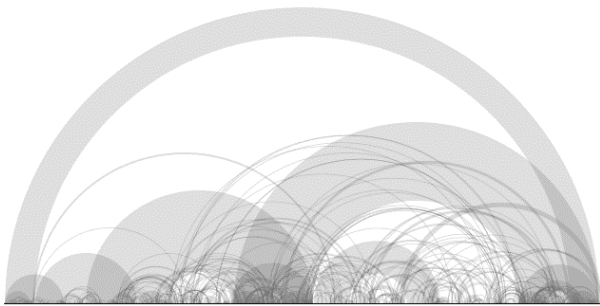
## 4. Applications to music

One of the most promising applications of arc diagrams is to reveal structure in musical compositions. An example of a musical arc diagram is shown in Figure 9, which represents the first line of the song *Mary Had a Little Lamb*.



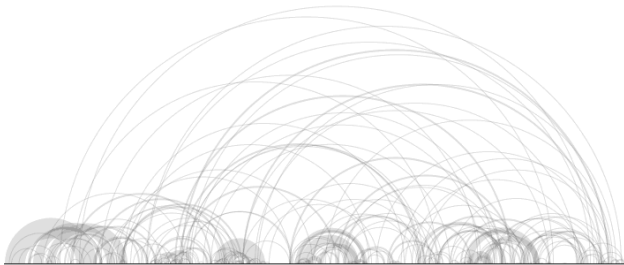
**Figure 9. Arc diagram for music**

Each arc connects two matching passages, where a "match" means that they contain the same sequence of pitches. The diagram shows repeated subsequences of three or more notes. To clarify the connection between the visualization and the song, I have displayed the score beneath the arcs.



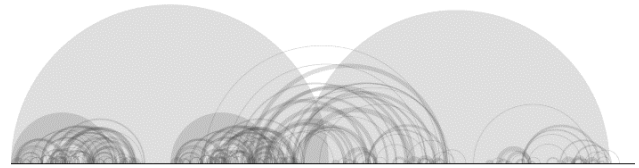
**Figure 10. Arc Diagram of *Für Elise***

Figure 10 visualizes Beethoven's *Für Elise*. (In this and subsequent diagrams, the source sequences are too long to display legibly.) Again, matches are based on equality of pitch; where chords occur we consider only the top note. Despite this extremely limited definition of musical similarity, the resulting matching diagram reveals an intricate and beautiful structure. The picture shows how the piece begins and ends with the same passage, while a longer version of that passage repeats throughout at increasing intervals. Equally illuminating is the long stretch in the second half of the piece where that passage is not repeated at all and the structure looks distinctly different, which corresponds well to what you hear when you listen to the music.



**Figure 11. Toreador, Carmen**

Not all pieces show as much large-scale repetition as *Für Elise*. For instance, the "Toreador" song from *Carmen* (Figure 11) looks completely different. Instead of a few long passages repeated over and over again, it contains many repeated smaller phrases.



**Figure 12. Minuet in G Major, Bach**

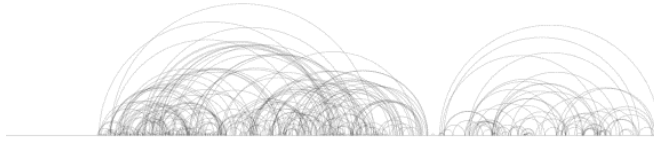
As a final example, consider Bach's *Minuet in G Major* (Figure 12). The arc diagram shows that the piece divides into two main parts, each made of a long passage played twice: what a musician would call an "AABB" structure. AABB is, in fact, the classic structure of a minuet, which shows that the matching algorithm is picking out structures that correspond to conventional musical analysis. The pictorial representation, however, provides much more detailed information than the simple "AABB" notation. For instance, you can see that the A and B passages are loosely related, as shown by the bundle of thin arcs connecting the two halves of the piece. And the fact that the two main arcs overlap shows that the end of the A passage is the same as B's beginning.

For musical compositions it is natural to consider the sequence of differences between successive notes as well as the notes themselves. Figure 16 (at the end of the paper) shows two arc diagrams for *Für Elise*, juxtaposed: the top is a large version of Figure 10 and the bottom, flipped diagram shows additional matching substrings based on intervals between successive notes.

## 5. Finding structure in text and compiled code

Arc diagrams are well suited to the analysis of highly structured data such as musical compositions, but they also can be effective in exploring other less well-structured data. Three examples we consider are compiled computer code, a web page, and a nucleotide sequence from DNA.

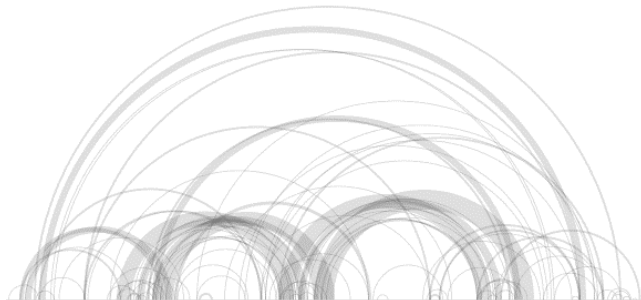
To see how matching diagrams can find structure in sequences that might otherwise be difficult to decipher, consider Figure 13, which shows the bytecode for the main Java class of the diagram-generating application itself:



**Figure 13. Java class file (bytecode)**

The diagram clearly shows that the file has two main sections. This reflects the important piece of structural information that Java class files are divided into two main sections: the constant pool and the executable code. Moreover, the diagram shows that the initial section (the constant pool) takes up significantly more memory than the code itself. Thus the diagram provides both structural and quantitative information that would be difficult to discern from a standard "hex-dump" text view of the file.

When arc diagrams are applied to textual data, they can also produce useful results. For example, an arc diagram of a short HTML file resulted in the picture in Figure 14.



**Figure 14. HTML page**

The page was organized into three basic sections, a fact which was delineated clearly by wide arcs. (These correspond to the HTML code for images and tables.) At the same time, the finer-grained detail is also revealing. For instance, the diagram shows that the last section of text has many connections to previous parts of the page, with especially strong connections to the beginning; this indicates that the introduction and conclusion of the text contained similar phrases and themes.

Finally, it is natural to apply matching diagrams to DNA nucleotide sequences. One potential pitfall is that DNA is noisy data in the sense that exact repetition on a large scale is uncommon due to mutations. In some situations,

however, this is not a problem. For example, there is significant interest in understanding patterns in upstream transcription regions (UTRs), i.e. the subsequences of DNA that precede regions that code for genes. The distribution of certain small (typically around 7 base pairs) subsequences called *motifs* in a gene's UTR is thought to play a key role in regulating that gene [C00].

Figure 15 shows an arc diagram for a UTR of length 500 for a particular yeast gene (identifier YGL123C in [S02]), filtered to show repeated patterns of 7 or more symbols. Although not as dramatic as the music diagrams, this picture does contain interesting information. For example, it shows that one special region of the UTR (from roughly 200 steps before the end to 100 steps before the end) contains at least one instance of most of the repeated patterns. This is potentially related to a recent finding [H00] that many regulatory motifs are more likely to appear in this same restricted region.



**Figure 15. DNA sequence**

## 6. Summary and directions for future work

Arc diagrams are a promising new method for visualizing sequences. They are well suited to displaying structure in sequences that contain complex patterns of repetition. We have shown examples of their potential use in domains ranging from text to DNA, although analysis of musical form is perhaps the most promising application.

Many areas remain for future exploration. One key direction of exploration is the best way to add interactivity to the diagrams. The visualizations described in this paper are static; an interactive version could be more powerful. One natural extension would be to add sliders to control the level of detail, allowing the user to specify how large a subsequence would need to be in order for an arc to be drawn. In addition, users could be allowed to drill down for details. For example, if the user pointed at a particular arc, the subsequence corresponding to that arc could be drawn on screen. If the underlying sequence were a musical composition, that particular passage could be played.

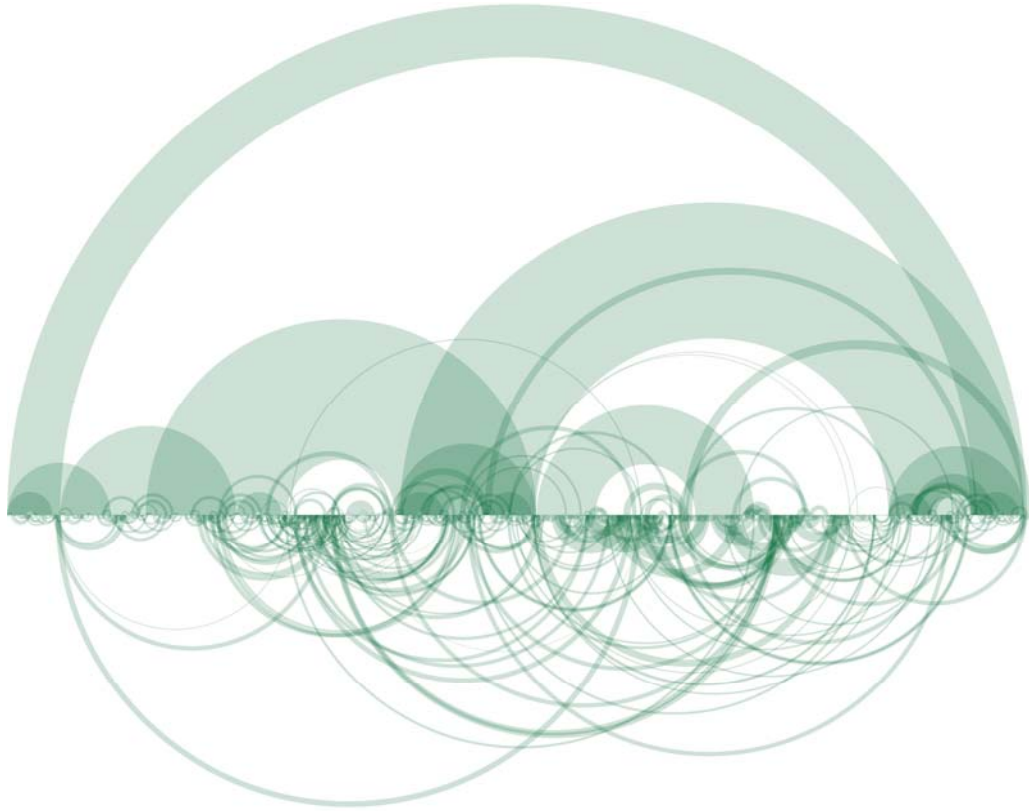
Another area for future investigation is the use of alternative pattern matching algorithms. Instead of drawing arcs between substrings that are identical, one

could choose a more flexible criterion. For example, when diagramming a fugue the criterion for a match might include transpositions and inversions as well as identical repetition. In addition, by using "fuzzy" matching techniques, it might be possible to make the method more useful for noisy data, such as DNA sequences.

A final area to explore is the incorporation of additional variables into the visualization. One might use different hues to indicate substrings that match according to different criteria. Another technique for adding information would be to incorporate a notion of intensity (e.g., corresponding to volume in a musical composition) and draw arcs with a translucency factor corresponding to the intensity.

## 7. References

- [C00] Cooper, Geoffrey. *The Cell: A Molecular Approach*, 2<sup>nd</sup> ed. Sinauer Assoc. 2000.
- [CH92] Church, K.W., and Helfman, J.I. "Dotplot: A Program for Exploring Self-Similarity in Millions of Lines of Text and Code", *Proceedings of the 24th Symposium on the Interface, Computing Science and Statistics V24*, pp. 58-67, March, 1992.
- [H00] Hughes, Jason *et al.*, "Computational identification of cis-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*." *Journal of Molecular Biology* (2000) 296: 1205-124.
- [HR83] E. Hamori and J. Ruskin, "H-Curves, a novel method of representation of nucleotide sequences especially suited for long DNA sequences," *Journal of Biological Chemistry*, 258 (2):1381-1327, 1983.
- [J90] H. J. Jeffrey. "Chaos game representation of gene structure." *Nucleic Acids Research*, 18(8):2163-2170, 1990.
- [JS95] D. Jerding and J. Stasko. "The Information Mural: A technique for displaying and navigating large information spaces." *IEEE Visualization '95 Symposium on Information Visualization*, pp 43-50.
- [S69] Schenker, Heinrich, *Five graphic music analyses*, New York: Dover, 1969
- [S02] *Saccharomyces genome database*, <http://genome-www.stanford.edu/Saccharomyces/> (2002)
- [T02] "TimeSketch." *ECS Media*. [www.ecsmedia.com](http://www.ecsmedia.com) 2002
- [W01] M. Wattenberg, "The Shape of Song." <http://www.turbulence.org/works/Song> (2001)
- [W93] D. Wu, J. Roberge, D. J. Cork, B. G. Nguyen and T. Grace, "Computer visualization of long genomic sequences." *IEEE Visualization '93*, pp. 308-315.



**Figure 16. *Für Elise*, exact (top) and modulated (bottom) matches**

# KeyStrokes: Personalizing Typed Text with Visualization

Petra Neumann, Annie Tat, Torre Zuk, and Sheelagh Carpendale

Department of Computer Science, University of Calgary, 2500 University Drive NW, Calgary, AB, Canada T2N 1N4



---

## Abstract

*With the ubiquity of typed text, the style and much of the personality of handwriting has been lost from general communication. To counter this we introduce an artistic real-time visualization of typed messages that additionally captures and encodes aspects of an individual's unique typing style. The potential of our system to augment electronic communication was evaluated and the results are provided along with analysis of their implications for social visualization.*

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [Information Interfaces and Presentation]: User Interfaces – Graphical user interfaces (GUI); I.3.3 Computer Graphics Display Algorithms

---

## 1. Introduction

Electronically written text communications are becoming the standard for today's correspondence. E-mail and instant messaging are already replacing handwritten letters and messages. Even e-cards are now being used for birthdays or holidays as a replacement for the physical card. People can converse across distances electronically quickly and cost-effectively, making it a very popular choice for conversation. However, typed text messages lack the personal character of handwriting. Some characteristics of the message author's writing style, such as neatness of writing, or how individual letters are shaped is lost in typed messages. This lack of personal character has led to attempts to enliven electronic messages through ASCII art, emoticons, or through the embedding of HTML options.

The goal of this work is to build visualizations that automatically encode personal typing characteristics to enrich communication. By looking at how people type an electronic message, we can notice many different typing styles involving typing speed, typing rhythm, hand-usage, and how many times letters or words are erased, reprinted, or replaced. We capture and use the details of a person's style to create a visual representation of a message that can then be used

for asynchronous distribution, for example, as an electronic postcard. Our visualization differs from previous approaches in that we focus on visualizing the process of creating a message whereas previous work has mostly been concerned with visualizing characteristics of the already created words and sentences.

This work has two main contributions. Our first contribution is the KeyStrokes system for visualizing personal and message characteristics of typed text. We know of no other information visualization that attempts to display this type of data for personalizing electronic communication. An evaluation and an analysis of the system in terms of its design and motivation forms the second contribution of this work.

## 2. Related Work

KeyStrokes is part of a growing body of research that uses text as a source for social data analysis. Text, in its various forms, is probably one of the most prevalent data sources available today. Thus, not surprisingly, a large number of visualization techniques have been developed that represent different aspects of textual data. The body of work most related to our system is concerned with visualizing the social aspects of text-based communications. Several visualiza-

tions of persistent conversations (conversational exchanges with applications such as e-mail, blogs, instant messaging, etc.) have explored ways to uncover the underlying social patterns. For example, The Babble System reveals social awareness of online chat activities through a social proxy visualization [ESK\*99]. ChatCircles [DKV98] shows synchronous conversation, visualizing one's presence, activity level, and chat identity. CrystalChat [TC06] integrates visual representations of social patterns with temporal aspects of chat conversations. It has been noted that observing graphical patterns of one's own communications encourages retrospection and story-telling [VBN\*04, TC06]. Perhaps inspired, as we have been, by the proliferation of emoticon use as evidence that people want to include their emotional state in their messages, there has also been research into visualizing emotion [TC06, LD06]. A visualization of emotional content of blog messages has been developed by using the words preceded by "I feel" and "I am feeling" [HK06]. In this vein, the work that most closely relates to our project is Cheiro [LD06], an animation of text that is based on mouse gestures. However, each typed word requires the user to gesture with the movement of the mouse.

Studies have shown that monitoring the intervals between keystrokes and duration of keystrokes as an individual types is sufficient to support the determination of their identity [She95]. From this, we know that it is possible to distinguish individual users' typing style by looking at these characteristics. Our research aims at embedding visuals representing one's unique typing characteristics within the typed message. One important advantage to this approach is that our visualization can be created without any extra effort on the part of the person typing the message.

### 3. KeyStrokes Visualization

We had several design goals in creating our visualization. Foremost, we wanted the visualization to minimize effort for the person typing the message. To do this, we extract keystroke data during typing and use it to create a responsive visualization so that the visuals representing a given key stroke would appear rapidly enough for the connection between action and response to be evident. We also wanted to create a visually appealing design that would be scalable for different sized screens. We use typing style and textual content to develop patterns to enrich and personalize a message. Our visualization currently uses the metaphor of a postcard that can be filled with our visualization of a message on one side and the typed text of the message on the other. In developing KeyStrokes, we considered design criteria such as background and foreground objects, splattering effects, and differing stroke styles including stroke movement and direction analogous to strokes created with a paint brush.

### 3.1. Visualizing Writing Patterns

Each letter of the alphabet and some common punctuation keys are represented at a fixed 2D spatial location in our visualization corresponding to a jittered physical English QWERTY keyboard layout (Figure 1).

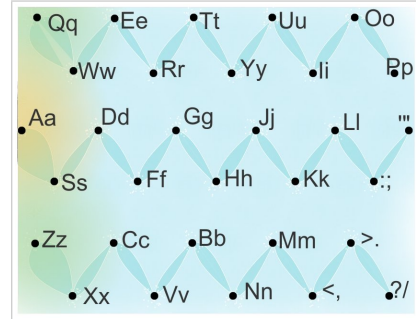
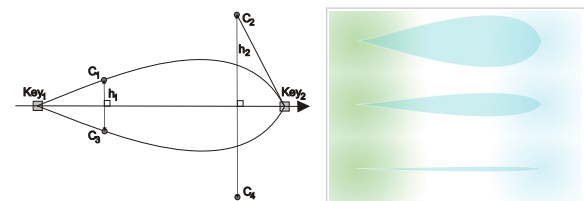


Figure 1: Mapping of key locations.

When a key sequence has been pressed, we connect the corresponding key locations with a semi-transparent stroke to mimic the strokes created with a brush or pen. Figure 2(a) gives an overview of the design of a stroke. The strokes are drawn with two Bézier curves using two control points on each side to give the stroke a visible direction from thin to thick. The height  $h$  of the control points  $c_i$  is determined by the amount of time between keypresses. In Figure 2(b) the top key combination was typed slowly resulting in a wide stroke. Compare this to the middle and bottom stroke where there was a much shorter delay between keypresses resulting in narrower strokes. In this way, the strokes connecting each sequential keypress implicitly reveal the temporal movement of fingers (and hands).



(a) Stroke design with Bézier control points and key locations. (b) Slow (top), medium (middle), and fast (bottom) strokes.

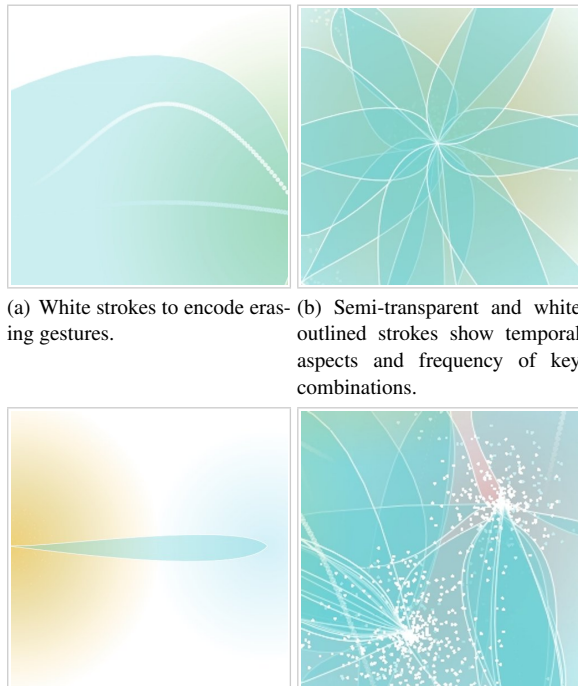
Figure 2: Stroke Design (a) and stroke types (b).

For many people, writing style can also be distinguished by how many times letters have been erased, retyped, or replaced. We show the use of backspacing between key combinations by a curved white line connecting the two keys while erasing the previously created stroke (see Figure 3(a)). Note the many backstroke lines in Figure 4 where an artistic placement of keystrokes has been attempted. The curved line is drawn to imitate a crossing-out motion in hand-written

text where mistakes are not completely erased even when an eraser or white-out is used.

### 3.2. Visualizing Message Patterns

One way visualized message patterns are shown in our system is through the frequency of letters and keystroke sequences. The frequency of pairwise key sequences becomes visible through the overlap of the semi-transparent and white outlined strokes, as can be seen in Figure 3(b). We encode several message characteristics in the background of the visualization. The frequency of an individual key is emphasized through a transparent circle in the background (see Figure 3(c)). When a key is more frequently pressed, the colour of the circle will change from blue to pink or cool to warm colours. To aid discrimination and comprehension, we redundantly encode repeated key presses with a splash of white dots around the key location, increasing the radius and spread of the splash after each key press. An example is given in Figure 3(d). Another characteristic that is visualized in the background is word beginnings. At the beginning of a word, vowels are drawn with a yellow background and consonants with a green background to visualize soft and hard sounds. The change in background colour is used to add dynamics and to balance the whole composition. Figure 4 shows all of the mentioned representations combined.



(a) White strokes to encode erasing gestures. (b) Semi-transparent and white outlined strokes show temporal aspects and frequency of key combinations. (c) Circles in the background encode message patterns. (d) White splashes encode frequencies of keypresses.

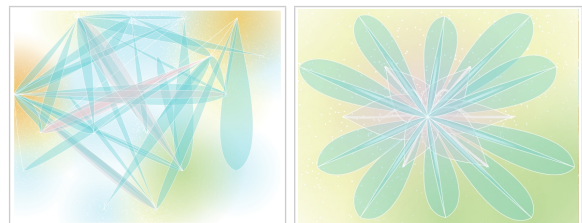
**Figure 3:** Visualization characteristics.



**Figure 4:** A painted message showing the combination of all message pattern representations.

### 3.3. Interaction

User interaction with our visualization is natural, requiring nothing beyond normal typing. As soon as one starts to type, the visualization space is filled with painted strokes in real-time and recently placed strokes are animated. The animation shows strokes vibrating in the display for a short period of time to enforce the dynamic nature of the visualization and to show where the last letter was typed on the screen. During informal demonstrations of the system in our lab, we noticed two very different usage patterns. Many people tended to compose a meaningful text that was conveyed in the visualization (Figure 5(a)). Others started to create intentional artwork after learning how and where keystrokes were displayed in the visualization. The typed words did not have any meaning attached to them, but the created image carried the message, as in Figure 5(b) where a floral pattern was created to send to a close friend.



(a) A message with meaningful text. (b) Message content embedded in the image.

**Figure 5:** Two different types of messages.

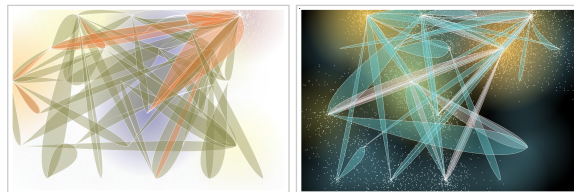
### 3.4. Individualization

The images shown at the top of the first page represent a visualization of a poem typed by four different people. Different writing styles and similarities in typing become apparent by how the strokes are printed in the visualization. It is possible to get an overall feel for the individual typing speeds,



with the third typist being generally slower. You can also see how different key combinations took longer for certain individuals to type. The individuals typed essentially the same message which can be recognized through the similar stroke pattern and by how background colours are placed. A common characteristic seems to be that all individuals seemed to pause before pressing the final character “.” This shows as the thicker stroke on the right side of each image.

Another way to personalize a KeyStrokes message is through the selection of different colour themes so that the tone or feeling of a message can be individually selected. Figure 6 shows two additional colour themes we developed.



(a) A slightly darker theme. (b) A pastel theme on black.

**Figure 6:** Two different colour themes that can be selected.

#### 4. Personalization with KeyStrokes—An Evaluation

After the initial design of the KeyStrokes software we received a number of positive responses from casual users in our research laboratory. To further assess the response to and effectiveness of our visualization design in a more general setting, we designed a questionnaire and collected responses during two demonstration sessions at international conferences. The results of this assessment indicate that the KeyStrokes visualization was well received and also raise several interesting points for discussion.

##### 4.1. Design of the Questionnaire

The questionnaire contained four types of questions: general background (demographics), information relating to the motivation for this work, information on the current visualization, and a general comment field concerning the KeyStrokes system. The background questions asked participants to state their occupation, age group, electronic communication use and frequency, and hand-writing frequency. Questions 1–4 were answered using a five-point Likert scale (strongly disagree (1)–strongly agree (5)).

Question 1 was specifically targeted at one of the main motivations for our work:

*Q1: I find electronic communication lacking in personality.*

Our hypothesis was that recipients would generally find electronic communication lacking in personality. With Question 2 we wanted to determine whether this lack of personality was generally seen as a benefit or drawback of electronic communication:

*Q2: I value the anonymity of electronic communication.*

Questions 3 and 4 asked about the main data source for our visualization and whether participants had any privacy concerns with the visualization of this data:

*Q3: Visualizing finger positions, key transition speeds, and editing can capture some of my character.*

*Q4: Visualizing finger positions, key transition speeds, and editing reveals too much of my character.*

The remainder of the questions asked about the motivations and usage patterns for the KeyStrokes system.

*Q5: Would you use KeyStrokes visualization to augment your communication? (yes/no)*

*Q6: Why?*

*Q7: When?*

*Q8: In conjunction with what type of electronic communication?*

##### 4.2. Study Setup

We collected responses to this visualization through a questionnaire given out in paper form at two conference demonstration sessions. At each of these sessions, we set up a laptop running our KeyStrokes system with an external keyboard. Each participant was introduced to the theoretical background of the system and its different functionalities. We encouraged participants to try the different features of the system and to ultimately type a message, thus creating a KeyStrokes visualization that we printed for them on 4" × 6" photo paper. During the printing, we asked the participants to volunteer to fill out our questionnaire. These four-hour demonstration sessions were held at the 2006 IEEE Symposium on Information Visualization (InfoVis'06) poster session [NTZC06a] and the 2006 ACM Conference on Computer Supported Cooperative Work (CSCW'06) demonstration and poster session [NTZC06b].

##### 4.3. Participants

A total of 68 people (37 InfoVis'06, 31 CSCW'06) completed our questionnaire. We included demographic questions to determine if answers were different according to age, occupation, gender or between the communities at the two conferences. However, we found no significant differences for any of these variables with the exception of the electronic communication use of e-cards which were reported to be sent/received significantly more by participants at CSCW'06 (2-sided Fisher's Exact Test,  $p = .034$ ), and these participants also reported significantly more electronic communication usage in the "other" category (2-sided Fisher's Exact Test,  $p = .035$ ); mostly video and VOIP services. Participants stated they most heavily used email (97% total), instant messaging (IM) (72% total), and text messaging (48.5%). Electronic communication was pervasive with more than 60% of our participants reporting that they used hand-written communication only "yearly" or "never anymore," while all of our participants reported to use electronic communication daily.

Question	Sex	Occupation	Age	Conference
	U	$\chi^2$ (df)	$\chi^2$ (df)	U
	p	p	p	p
1	106	4.71(4)	5.729(4)	497.5
	.814	.324	.221	.336
2	74	2.266(4)	3.876(4)	497
	.310	.696	.430	.6
3	94	2.567(4)	4.03(4)	543.5
	.667	.675	.414	.886
4	81.0	1.927(4)	2.285(4)	553.0
	.301	.763	.684	.983

**Table 1:** We found no significant correlation between the answers to Q1–4 and the background variables sex, occupation, age, and conference. Scores are reported according to two-tailed Mann-Whitney Test (Sex, Conference) and Kruskal-Wallis Tests (Occupation, Age).

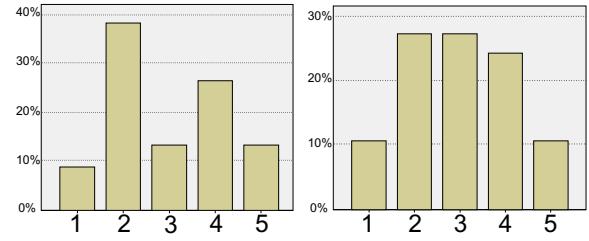
#### 4.4. Analysis Method

For the analysis of relationships between all collected variables in the questionnaire, the threshold for statistical significance was set at  $p < .05$ . For categorical data we used Pearson’s Chi-Square measure when less than 10% of reported frequencies had a count of  $< 5$  and Fisher’s Exact Test for small sample sizes. For ordinal data we used the Mann Whitney test for two independent samples and the Kruskal Wallis Test for  $k$  independent samples. Due to the ordinal nature of our variables and also the relatively small sample size we used non-parametric tests to determine relationships between specific variables. We determined whether there was a correlation between questions by doing a pairwise comparison of the answers to the questions by using the appropriate above-mentioned tests.

#### 4.5. Results

Results will be provided with interpretations to follow in Section 5. For Questions 1–4 we found no significant difference between the respective responses and the demographic variables sex, occupation, age, and conference through pairwise comparison (see Table 1). Overall, participants reported to either agree or disagree on whether they found electronic communication lacking in personality (Q1). 47% of participants disagreed or strongly disagreed and 40% agreed or strongly agreed with this statement. Figure 7(a) gives a graphical overview of the bimodal distribution of answers to this question. Figure 7(b), the responses to Q2, show that participants did not have a consensus on whether they valued the anonymity of electronic communication. 38% of participants disagreed, 35% agreed, and 37% were undecided.

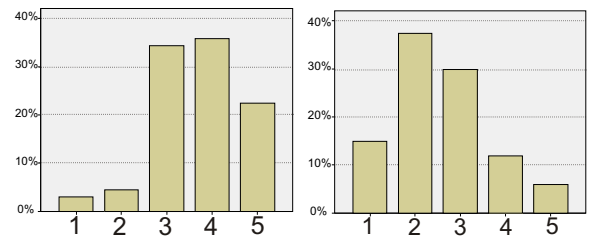
In Question 3, participants tended to agree that visualizing finger positions, key transition speed, and editing could capture some of their character. Participants also generally disagreed in Question 4 that the visualization of this data would reveal too much of their character. Figure 8 gives



(a) Q1 “I find electronic communication lacking in personality”. (b) Q2 “I value the anonymity of electronic communication”.

**Figure 7:** Answer frequencies to Questions 1 and 2.

an overview of the answers to these two questions. Overall, 71% of people reported in Question 5 that they would use KeyStrokes to augment their communication. Two recipients (3%) did not report either yes or no and wrote a “maybe” next to the provided checkboxes.



(a) Q3 “Visualizing finger positions, key transitions speed, and editing can capture some of my character”. (b) Q4 “Visualizing finger positions, key transitions speed, and editing reveals too much of my character”.

**Figure 8:** Answer frequencies to Questions 3 and 4.

Questions 6–8 were free-form questions. We combined similar answers to each question into distinct categories for analysis. For example, the following answers to Question 6: “It’s personal”, “To personalize my email, blog”, “for personal notes” were combined in a category called “personalization”. Question 6 “Why?” was asked in direct reference to the answer given in Question 5 “Would you use KeyStrokes to augment your communication?”. Table 2 gives an overview to the main categories of answers given to Question 6 in relation to Question 5.

There were 16 different answer categories for Question 7 (“When?”—see Table 3). 39 people answered this question. Only three people who reported that they would not use the tool gave an answer to this question: personal correspondence (2), e-mail (1). Table 4 gives an overview of answers to Question 8: “In conjunction with what type of electronic communication?”. In the general comments field participants gave mostly appreciative comments and advice about how to improve the tool. We will report on those comments in more detail in the following discussion.

Would use the tool			
Yes		No	
Why	# (%)	Why	# (%)
Personalization	20(46.5)	Insufficient information	5(29.4)
Fun	11(25.6)	Too confusing	3(17.6)
Visually Appealing	7(16.3)	Can't envision usage	3(17.6)
It's novel	2(4.7)	Not for professional communication	2(11.8)
Speed	1(2.3)	Fun	1(5.9)
Like the idea	1(2.3)	It depends	1(5.9)
Can't envision usage	1(2.3)	Too distracting	1(5.9)
It depends	1(2.3)	Lack of control	1(5.9)
	43(100)		17(100)

**Table 2:** Q6 (“Why”) in relation to Q5 (“Would you use KeyStrokes to augment your communication?”).

	When	# (%)
	Personal Correspondence	19(48.7)
	E-mail	3(7.7)
	Not Sure, Daily, Greeting Cards	2(5.1)
	Greeting Cards & IM, Greeting Cards & Electronic Signature, IM, Learning to Type, Whenever it is ready, To impress, Any text, When time permits, For fun, Correspondence to art-minded colleagues, Occasionally	1(2.6)

**Table 3:** Answers to Question 7: “When?”.

### 5. Interpretation of the Results

From the questionnaire we learned that over 70% of our participants said they would use the tool. This can be seen as a success for an information visualization tool that participants had only experienced for a few minutes during our demonstration sessions. However, through our sampling method participants were self-selected and obviously in some way interested in the tool by attending our demonstration. Nevertheless, examining the results of our questionnaire gave us interesting feedback on the tool, our motivation, design, and future work. The following sections contain more detail about our interpretations of the data and hypothesize on why the KeyStrokes tool received such positive responses. For the interpretation of each of the questions, it is important to keep in mind that all answers were given by participants from the

Electronic Communication	# (%)
Email	33(48.5)
IM	16(23.5)
Ecards	5(7.4)
Any Text	3(4.4)
Blogs	2(2.9)
SMS, Wiki	1(1.5)

**Table 4:** Answers to Q8—“In conjunction with what type of electronic communication?”.

visualization creation standpoint, we did not ask participants to read and interpret messages that other people had created.

#### 5.1. Is Personalization a Motivation to use KeyStrokes?

Participants reported their main motivations to use the tool were personalization, fun, and visual appeal (Table 2). Personalization was actually also one of our main motivations to design the tool. We saw a general lack of personal characteristics in electronically written communication and set out to design the tool to bring personality back into electronic communication. In the questionnaire we asked whether participants agreed with this motivation. We found that participants responses were quite dispersed as to whether they found electronic communication lacking in personality or not. Interestingly, however, 21 of 32 people who did not find electronic communication lacking in personality reported that they would use the tool, even though they did not agree with this motivation. Ambiguity in the question might also have allowed responses relating to personality in the message rather than the medium. While a significant proportion of responses did agree with the motivation for the work, it is unknown if those who disagreed have very different measures of personalization. A question to provide a baseline reference might have been “Do you feel hand-written messages have more personality than electronic messages?”

#### 5.2. What Makes KeyStrokes “fun” to Use?

A quarter of the participants who said they would use the tool reported “fun” as their main motivation. This characteristic is not one commonly reported of information visualization tools. KeyStrokes includes some game-like features, as reported in [Mal80], that could lead to participants saying it was “fun” to use:

- The tool is challenging. It has a main goal: to see or communicate personal typing and message characteristics. It also has an uncertain outcome: typing characteristics are hard to foresee with changing messages and are also different between people and many of the encoded variables are hidden for users to discover.
- The tool has an emotional aspect to it. One can learn about one’s own typing patterns and also share this personal information in a message. One of our participants commented: “This adds a loving touch to notes.”
- The tool evokes curiosity: the tool attracts attention through its visual appeal (as reported by a number of participants, see Table 2) and pulsating strokes that indicate recently pressed key combinations. It engages people in interpreting the visualization and its novelty encourages people to explore it more.
- The tool encourages creativity: we observed people circumventing the intended usage of the tool to create interesting looking patterns (e.g. floral patterns) as the main content of a message (Fig. 5). Some people thought of

very creative ways to use our tool that we had not previously considered: for learning to type, electronic signatures, SMS, blogs, or wikis (Table 3 and 4).

- The tool is easy to use: one common characteristic of popular games is that they are quite easy to learn or provide appropriate help for learning to play the game. A KeyStrokes visualization can be created without much effort while typing a message and can then be attached to the message to share with others. The design of visualizations that require minimal effort to use is an important venue to consider in the area of information visualization.

### 5.3. What Negative Aspects were Reported?

Despite the majority of positive responses about the tool, about 30% of all participants reported that they would not use the tool, or at least not in its current form. Their main motivations were: a lack of information in the visualization, the visualization being too confusing, or not being able to imagine a use for the tool (see Table 2). During the demonstrations many users of our system expressed that they wanted to read the actual content of the message from the visualization in conjunction with getting an overview of the patterns of the message and the typing characteristics of the composer. Therefore, the first motivation may be related to the second one in that people found the visualization too confusing because they could not read the actual message back from the graphic.

### 5.4. Are there Privacy Concerns?

The questionnaire data generally confirmed our choice of typing characteristics used for the KeyStrokes visualization. Overall participants agreed that visualizing finger positions, key transition speeds, and editing habits could capture some of their character (see Figure 8(a)). One of our concerns while designing the visualization was that people would have privacy concerns and would, for example, not like to be identified by someone else as a slow typer or as someone who made lots of mistakes while typing. Generally, participants did not confirm this concern (see Figure 8(b)). However, a quarter of those participants who supported that KeyStrokes could capture some of their character also affirmed that it would reveal too much of their character. So overall, we did identify some privacy concerns among participants. This raises an interesting point for the field of information visualization, as often the goal of a visualization is to reveal as much information as effectively as possible. Our tool, however, can capture and visualize more data than some users might want to share with others.

### 5.5. Did Participants Like the Aesthetics?

It has been shown that the use of aesthetics and visual abstraction as part of the visualization can attract people's attention and interest [Tra97, Nor02]. We deliberately tried to

create visual mappings of typing characteristics with abstract and aesthetically appealing graphical representations. In the questionnaire we received overall positive responses for our visual design. In fact, a quarter of participants who would use the tool reported its visual appeal as the main motivation. This also confirms the above mentioned findings by Norman and Tractinsky [Tra97, Nor02]. Several participants also gave positive feedback on the design in the general comments field (e.g. "It's beautiful work", "Thank you for the beautiful e-card", etc.). Some participants requested changeable colours, and stroke control, or to use it as a visualization of currently typed text rather than a visualization of the complete message. Colour and stroke control will enable users to set the "tone" of the visual message enabling a more direct display of the moods and feelings the sender had when typing a message or even parts of a message.

### 5.6. Where Can the Tool be Used?

During the design, we envisioned KeyStrokes to be used in an electronic communication environment like an email or chat client. During our demonstration sessions we had deliberately not embedded the tool into such an environment in order not to restrict the users in their answers to Question 8. The main envisioned usage by our participants corresponded to ours. However, we received several interesting application ideas from participants, in particular, to use it for cell-phone text messages or in an email subject line. We believe that our principle design idea is scalable and can be adapted to small screens and display areas. We will consider these ideas for future versions of our tool.

### 5.7. KeyStrokes as a Social Data Analysis Tool

Wattenberg describes several hypotheses for the popularity of the online NameVoyager tool in [Wat05]. He hypothesizes that its popularity stems from the tool being part of an online social environment. Similar to our tool, he also suggests that his tool has game-like features that make it fun to use and suitable for social data analysis. In his paper he defines social data analysis as "a version of exploratory data analysis that relies on social interaction as source of inspiration and motivation." This definition seems to apply to our tool as well. KeyStrokes was built with the intention to share information visualizations with others making it essentially a social data analysis tool. One of our participants specifically confirmed this design in the open comments field: "A lot of fun to use, especially in the group setting." Wattenberg suggests that viewing exploratory data analysis as a social activity could explain much of the positive reaction towards his tool. We hypothesize this to be true for our tool as well but within a much closer community, in which the individuals know each other's character to some degree already. This hypothesis stems from the fact that many participants reported that they would use it for personalization when corresponding to friends and family or would not use

it for professional communication. The common ground of data analysis through our tool would be an understanding of the senders' character and typing skills at a certain point in time that could be read back and interpreted from the visualization. How the tool is used and accepted in the group setting when embedded in a specific communication environment will have to be determined in further evaluations.

### 5.8. Directions for Future Work

Results from our study suggest several directions for future work on KeyStrokes. One important aspect of the tool will be to further research its privacy implications. We would like to examine which types of information would make participants most uncomfortable if shared with others. Also, how such information can be hidden or transformed to make it more ambiguous needs further attention. In the field of CSCW several solutions to the problem have been explored for example in the area of screen sharing or video media spaces. These solutions include blurring or pixelating information that is often transmitted as pixel graphics. How or if these techniques can be applied to information visualizations and the KeyStrokes system in particular will have to be explored. In terms of the design of the visualization, we will add features to select colour or manipulate the principal stroke shape. Also, we would like to add the possibility of temporal reading of the strokes so that the actual letters of the message can be read back in order. With these changes, we will address the main points of critique uncovered during our study. An interesting and as yet unexplored venue for future work will include further studies on whether the tool can be used as an electronic signature. Previous work has shown that statistically users could be identified by how they typed their passwords [She95]. It seems possible that visualizations of this data could be used as electronic signatures.

### 6. Conclusion

The KeyStrokes system is a tool designed to enrich typed communication with personal characteristics. In this sense KeyStrokes is a social data analysis tool that allows shared analysis and exploration of personal data. The creation of this visualization was motivated by the lack of personal characteristics of electronic textual conversation compared to hand-written messages. KeyStrokes was created with several design goals in mind: to minimize the effort required to create and share the visualization, to encourage use of the tool through a visual appealing design, and to encode personal typing and message characteristics to bring character back into electronic communication. In order to assess the response and effectiveness of KeyStrokes, we performed a user study. The KeyStrokes tool received an overall positive response during our study, with many requests to make the tool publicly available. We identified several possible reasons for this positive response, discussed reported critique

of the system, and talked about feedback on our design, tool usage, and directions for future work. In general, we found that many participants felt electronic communication to be lacking in personality; so, visualizations that are built to aid in personalization fill a needed gap.

### Acknowledgements

We would like to thank Ilab members for useful comments and suggestions, Dr. Tak Shing Fung for his advice with the statistical analysis, and our funding agencies Alberta Ingenuity, iCORE, NSERC, and Veritas DGC Inc.

### References

- [DKV98] DONATH J. S., KARAHALIOS K., VIÉGAS F. B.: Visualizing Conversations. In *Proc. of System Sciences* (Los Alamitos, USA, 1998), IEEE Press.
- [ESK\*99] ERICKSON T., SMITH D. N., KELLOGG W. A., LAFF M., RICHARDS J. T., BRADNER E.: Socially Translucent Systems: Social Proxies, Persistent Conversation, and the Design of 'Babble'. In *Proc. of CHI* (New York, USA, 1999), ACM Press, pp. 72–79.
- [HK06] HARRIS J., KAMVAR S.: We Feel Fine. Website <http://www.wefeelfine.org/>, Accessed December, 2006.
- [LD06] LAM F., DONATH J.: Anthropomorphic Typography. In *Proc. of CHI Workshop on Social Visualization* (2006).
- [Mal80] MALONE T. W.: What Makes Things Fun to Learn? Heuristics for Designing Instructional Computer Games. In *Symposium on Small Systems* (New York, USA, 1980), ACM Press, pp. 162–169.
- [Nor02] NORMAN D. A.: Emotion & Design: Attractive Things Work Better. *Interactions* 9, 4 (July 2002), 36–42.
- [NTZC06a] NEUMANN P., TAT A., ZUK T., CARPENDALE S.: Interactive Poster: Personalizing Typed Text Through Visualization. In *Proc. Compendium of InfoVis* (Los Alamitos, USA, 2006), IEEE Computer Society, pp. 138–139.
- [NTZC06b] NEUMANN P., TAT A., ZUK T., CARPENDALE S.: Visualization of Typed Communication. In *Extended Abstracts and Interactive Demos of CSCW* (New York, USA, 2006), ACM Press, pp. 139–140.
- [She95] SHEPHERD S. J.: Continuous Authentication By Analysis of Keyboard Typing Characteristics. In *European Convention on Security and Detection* (1995), pp. 111–114.
- [TC06] TAT A., CARPENDALE S.: Crystal Chat: Visualizing Personal Chat History. In *Proceedings of HICSS* (2006).
- [Tra97] TRACTINSKY N.: Aesthetics and Apparent Usability: Empirically Assessing Cultural and Methodological Issues. In *Proc. of CHI* (New York, USA, 1997), ACM Press, pp. 115–122.
- [VBN\*04] VIÉGAS F. B., BOYD D., NGUYEN D. H., POTTER J., DONATH J.: Digital Artifacts for Remembering and Storytelling: PostHistory and Social Network Fragments. In *Proc. of HICSS* (2004), pp. 105–111.
- [Wat05] WATTENBERG M.: Baby Names, Visualization, and Social Data Analysis. In *Proc. of IEEE InfoVis* (Los Alamitos, USA, 2005), IEEE Computer Society, pp. 1–7.

# Discovering interesting usage patterns in text collections: Integrating text mining with visualization

Anthony Don<sup>1</sup>, Elena Zheleva<sup>2</sup>, Machon Gregory<sup>2</sup>, Sureyya Tarkan<sup>2</sup>, Loretta Auvil<sup>4</sup>, Tanya Clement<sup>3</sup>,  
Ben Shneiderman<sup>1,2</sup> and Catherine Plaisant<sup>1</sup>

<sup>1</sup>Human Computer Interaction Lab  
<sup>2</sup>Computer Science Department  
<sup>3</sup>English Department  
University of Maryland, USA

<sup>4</sup>National Center for Supercomputing Applications,  
University of Illinois, USA

{don,elena,mbg,sureyya,ben,plaisant} @cs.umd.edu, lauvil@ncsa.uiuc.edu, tclement@wam.umd.edu

## ABSTRACT

*This paper addresses the problem of making text mining results more comprehensible to humanities scholars, journalists, intelligence analysts, and other researchers, in order to support the analysis of text collections. Our system, FeatureLens, visualizes a text collection at several levels of granularity and enables users to explore interesting text patterns. The current implementation focuses on frequent itemsets of n-grams, as they capture the repetition of exact or similar expressions in the collection. Users can find meaningful co-occurrences of text patterns by visualizing them within and across documents in the collection. This also permits users to identify the temporal evolution of usage such as increasing, decreasing or sudden appearance of text patterns. The interface could be used to explore other text features as well. Initial studies suggest that FeatureLens helped a literary scholar and 8 users generate new hypotheses and interesting insights using 2 text collections.*

**Keywords:** D.2.14.a User interfaces, H.2.8.h Interactive data exploration and discovery, H.2.8.l Text mining

## 1. INTRODUCTION

Critical interpretation of literary works is difficult. With the development of digital libraries, researchers can easily search and retrieve large bodies of texts, images and multimedia materials online for their research. Those archives provide the raw material but researchers still need to rely on their notes, files and their own memories to find “interesting” facts that will support or contradict existing hypotheses. In the fields of the Humanities, computers are essentially used to access to text documents but rarely to support their interpretation and the development of new hypotheses.

Some recent works [4, 11] addressed this problem. One approach, supports the analysis of large bodies of texts by interaction techniques together with a meaningful visualization of the text annotations. For example Compus [4] supports the process of finding patterns and exceptions in a corpus of historical document by visualizing the XML tag annotations. The system supports filtering with dynamic queries on the attributes and analysis using XSLT transformations of the documents. Another approach is to use data-mining or machine learning algorithms integrated with visual interfaces so that non-specialists can derive benefit from these algorithms. This approach has been successfully applied in the literature domain in one of our prior project [11]. Literary scholars could use a Naive Bayesian classifier to determine which letters of Emily Dickinson's correspondence contained erotic

content. It gave users some insights into the vocabulary used in the letters.

While the ability to search for keywords or phrases in a collection is now widespread such search only marginally supports discovery because the user has to decide on the words to look for. On the other hand, text mining results can suggest “interesting” patterns to look at, and the user can then accept or reject these patterns as interesting. Unfortunately text mining algorithms typically return large number of patterns which are difficult to interpret out of context. This paper describes FeatureLens, a system designed to fill a gap by allowing users to interpret the results of the text mining thru visual exploration of the patterns in the text. Interactivity facilitates the sorting out of unimportant information and speeds up the task of analysis of large body of text which would otherwise be overwhelming or even impossible [13].

FeatureLens<sup>1</sup> aims at integrating a set of text mining and visualization functionalities into a powerful tool, which provokes new insights and discoveries. It supports discovery by combining the following tasks: getting an overview of the whole text collection, sorting frequent patterns by frequency or length, searching for multi-word patterns with gaps, comparing and contrasting the characteristics of different text patterns, showing patterns in the context of the text where they appear, seeing their distributions in different levels of granularity, i.e. across collections or documents. Available text mining tools show the repetitions of single words within a text, but they miss the support for one or more of the aforementioned tasks, which limits their usefulness and efficiency.

We start by describing the literary analysis problem that motivated our work and review the related work. We then describe the interface, the text mining algorithms, and the overall system architecture. Finally we present several examples of use with 3 collections and discuss the results of our pilot user studies.

## 2. MOTIVATION

This work started with a literary problem brought by a doctoral student from the English department at the University of

<sup>1</sup> A video and an online demonstration are available from <http://www.cs.umd.edu/hcil/textvis/featurelens/>

Maryland. Her work deals with the study of *The Making of Americans* by Gertrude Stein. The book consists of 517,207 words, but only 5,329 unique words. In comparison, *Moby Dick* consists of only 220,254 words but 14,512 of those words are unique. The author's extensive use of repetitions (Figure 1) makes *The Making of Americans* one of the most difficult books to read and interpret. Literature scholars are developing hypotheses on the purpose of these repetitions and their interpretation.

Everyone then sometime is a whole one to me, everyone then sometimes is a whole one in me, some of these do not for long times make a whole one to me inside me. Some of them are a whole one in me and then they go to pieces again inside me, repeating comes out of them as pieces to me, pieces of a whole one that only sometimes is a whole one in me.

Paragraph 1225, *The Making of Americans*

Figure 1: Extract from *The Making of Americans*.

Recent critics have attempted to aid interpretation by charting the correspondence between structures of repetition and the novel's discussion of identity and representation. Yet, the use of repetition in *The Making of Americans* is far more complicated than manual practices or traditional word-analysis could indicate. The text's large size (almost 900 pages and 3183 paragraphs), its particular philosophical trajectory, and its complex patterns of repetition make it a useful case study for analyzing the interplay between the development of text mining tools and the way scholars develop their hypotheses in interpreting literary texts in general.

This collaboration between computer scientists and humanity scholars is part of the MONK project ([www.monkproject.org](http://www.monkproject.org)), which brings together multidisciplinary teams from six institutions. Because this case study used a very unusual text we also tested FeatureLens with other collections: a technical book, a collection of research abstracts, and a collection of presidential addresses which we use here to describe the interface and also used in our pilot user study.

### 3. RELATED WORK

Visualizations have been applied successfully to retrieving, comparing, and ranking whole text documents [14, 16] and computer programs [3, 7]. Instead of ranking documents according to their content, FeatureLens ranks text patterns according to their length and frequency, and it provides a visualization of the text collection at the document level and at the paragraph level. These two levels of granularity allow the user to identify meaningful trends in the usage of text patterns across the collection. It also enables the analysis of the different contexts in which the patterns occur.

A recent interactive NY Times display [8] shows the natural representation of the text of the *State of the Union Addresses* with line, paragraph, and year categorization. It displays word frequency, location, and distribution information in a very simple manner which seemed to be readily understandable by the literary

scholars we have been interacting with. It allows search but does not suggest words or patterns that might be interesting to explore. It also does not support Boolean queries.

Visualizing patterns in text is also related to visualizing repetitions in sequences. A number of techniques such as arc diagrams, repeat graphs and dot plots have been developed and applied to biological sequence analysis [2, 5, 6]. Compared to DNA, literary text has different structural and semantic properties such as division into documents, paragraphs, sentences, and parts of speech that one could use to create a more meaningful visualization. Arc diagrams have been used to visualize musical works and text, and have advantages over dot plots [15], though it has not been shown how they can be adapted to large collections of text without creating clutter. TextArc [9] is a related project, which visualizes text by placing it sequentially in an arc and allowing a user to select words interactively and to see where in the text they appear. It does not support ranking of patterns and selecting longer sequences of words. Most of the tools describe above only handle small datasets and display the collection as a fixed level of granularity.

### 4. FEATURELENS

Figure 2 shows the graphical user interface of FeatureLens. The *State of the Union Addresses* collection consists of eight documents, one for each of President Bush's eight annual speeches (there were two in 2001 because of 9/11). The documents are represented in the *Document Overview* panel. Each rectangular area represents one speech and its header contains the title of the document, i.e. the year of the speech in this case. Within the rectangular representation of the document, each colored line represents a paragraph in this collection. When the document is very large each line may represent a unit of text longer than a paragraph so that the overview remains compact. FeatureLens computes the default unit of text to be such that the overview fits on the screen, and users can change that value using a control panel. For simplicity we call that arbitrary small unit of text a paragraph in the rest of the paper.

The *Frequent Patterns* panel, located on the left of the screen, displays the pre-computed text patterns generated by the data mining algorithms. Currently we combine only 2 types of patterns: frequent words, and frequent itemsets of n-grams (which capture the repetition of exact or similar expressions in the collection - more details in section 5). Frequent words naturally occur at the top of the pattern list since the default ordering of the list is by frequency. This also makes it easier for users to learn the interface with simple patterns, then move on to more complex patterns later on as they chose other sorting and filtering options.

In Figure 2, the list of patterns has been sorted by decreasing frequency and the user has clicked on four of the most frequent patterns. The location of the patterns is displayed on the *Document Overview*. Each pattern has been assigned a different color reflected in the *Legend* panel. When a paragraph contains one of the selected patterns, the color saturation of the line reflects the score of the pattern in the paragraph: the more occurrences of the pattern in the paragraph, the more saturated the color. I AM HERE

The *Collection Overview* panel shows a graph of the distribution of the support for each selected pattern. The vertical axis

represents the support of the pattern per document and the horizontal axis shows the documents of the collection. When the user lets the mouse hover on a specific portion of the graph a popup shows the exact number of occurrences. In Figure 2, the distribution of the word “world” is displayed in blue, showing that the word was barely used in the first speech.

By looking for lines that contain all the colors in the *Documents Overview*, it is possible to identify the parts of the text where selected patterns occur together. Clicking on a colored line in the overview displays the text of the corresponding paragraph in the *Text View* along with five paragraphs before and after the selection, to provide context while maintaining fast response time. In Figure 2, the user has selected a paragraph that contains all 4 patterns. A blue horizontal bar indicates which paragraph is currently displayed in the *Text View*. The text of the selected paragraph has a matching light-blue background. In the right margin of the *Text View*, small colored tick marks indicate the position of the occurrences of the patterns with respect to the scrollbar to make them easier to find. The occurrences of the patterns in the text are highlighted in color as well, matching the colors used in the overview and the legend.

The *Frequent Patterns* panel on the left provides many controls to search, filter and sort the list of patterns. A search box allows users to find patterns that include particular keywords or Boolean combinations of keywords. Patterns can be filtered by minimum size (i.e. number of words) and minimum frequency within the whole collection. Patterns can be sorted by length or frequency. Above the list of patterns, a check box allows users to append patterns to the current display in order to compare and study correlations between different patterns (the default option is to show one pattern at a time). Buttons allow users to load the history of previously explored patterns, load another collection, or set options such as the size of the text to be represented as a line in the *Document Overview*.

In the next section, we describe the pattern mining process used in FeatureLens, to explain how patterns other than the trivial single word patterns are mined from the text.

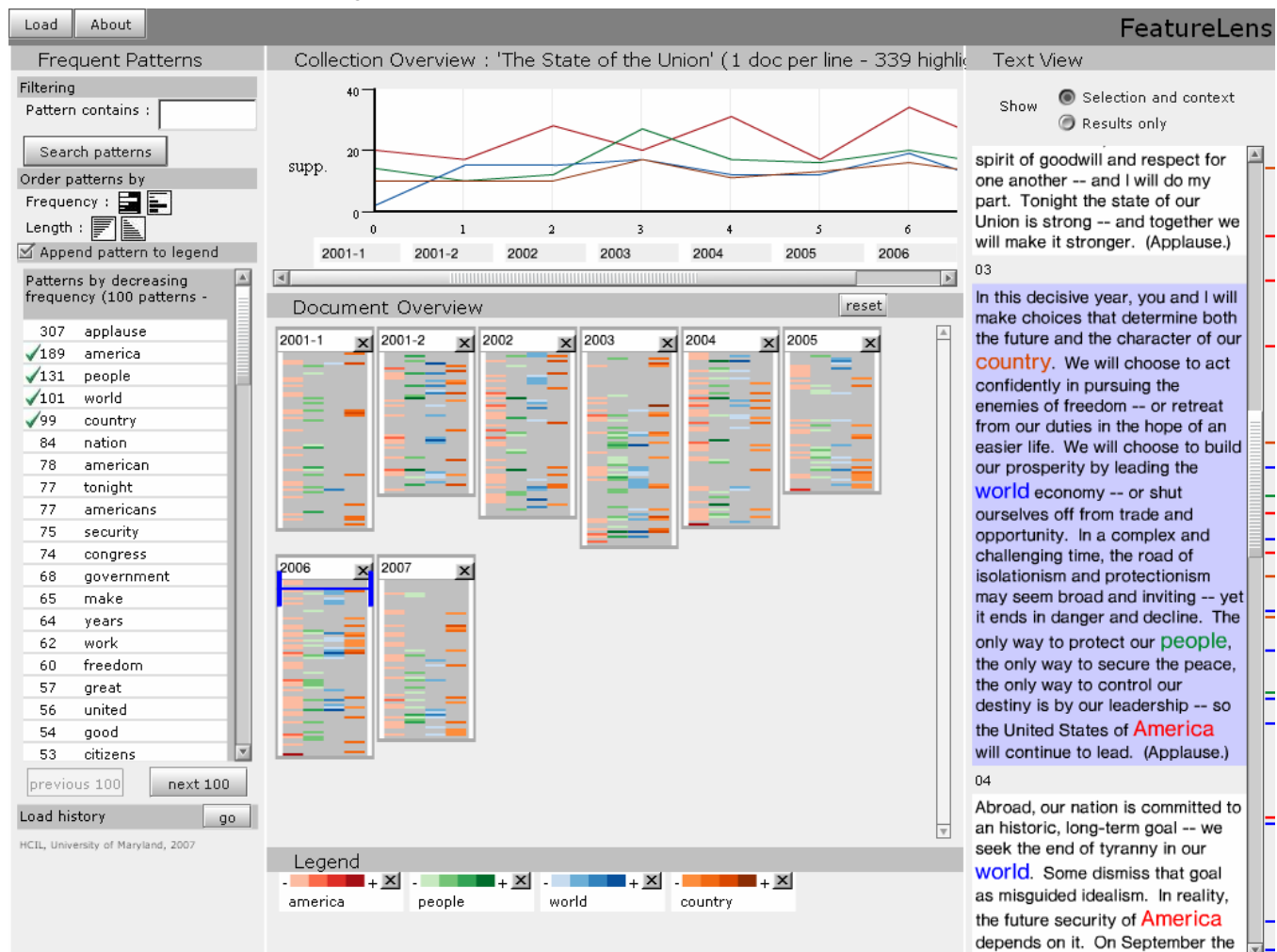


Figure 2: Main screen of FeatureLens with four of the most frequent text-patterns displayed in the overview.



## 5. MINING FREQUENT PATTERNS

For a given collection of texts, a hierarchical structure with two levels is assumed. Each collection contains documents which contain at least one paragraph (our chosen name for a small unit of text in this paper). This document-paragraph hierarchy can be used for a variety of text collections (see Section 7).

At this stage of the project, our focus is on the study of repetitions so we chose mining techniques that look for frequently occurring patterns, but we believe that the interface we developed can be used to interpret the results of other types of data mining techniques that generates lists of patterns.

Single words are the simplest form of patterns. For longer expressions, exact repetitions are useful because they often correspond to meaningful concepts or slogans, for example, “the No Child Left Behind Act” appears several times in President Bush’s speeches. Exact repetitions, though, cannot capture language constructions that include some varying parts, such as the ones in “improve our health care system” and “improve the health of our citizens.” In order to enable the user to study exact repetitions as well as repetitions with some slight variations, we resorted to the analysis of frequent closed itemsets of n-grams.

For each collection of texts, one set of frequent words and one set of frequent closed patterns of 3-grams are extracted using algorithms implemented in the Data-to-Knowledge (D2K) framework which leverages the Text-to-Knowledge (T2K) components [12].

### *Frequent expressions*

In order to qualify a word or a longer expression as “frequent,” we introduce the definitions of n-gram and the support of a pattern.

**Definition 1. N-gram:** a subsequence of n consecutive words from a sequence of words.

**Definition 2. Support of an expression:** Let  $C = \{p_1, \dots, p_n\}$  be a collection of  $n$  paragraphs, and let  $e$  be a text expression. The support of  $e$  in the collection  $C$ , denoted  $S(e, C)$ , is:

$$S(e, C) = \text{Cardinality}(\{p_i | e \subset p_i\}).$$

We consider an expression as “frequent” if its support is strictly greater than one. In case of large collections of texts, the threshold for the support may be increased in order to limit the number of frequent patterns.

### *Frequent words*

D2K/T2K provides the means to perform the frequent words analysis with stemming and we know that humanists are interested in looking at both stemmed and non-stemmed versions. We also know that sometimes, the humanist is interested in keeping stop words. In our current scenario, we did not use stemming, but we did remove stop words, such as 'a, 'the,' 'of,' etc. using the predefined list provided with T2K. One set of frequent words per collection of documents was computed using a minimum support of 1.

### *Frequent closed itemsets of n-grams*

Frequent pattern mining plays an important role in data and text mining. One relevant example is detecting associations between fields in database tables [1, 10]. We use these ideas in our text pattern analysis.

In order to provide repeated expressions that are exact repetitions as well as repetitions with slight variations, we propose to use frequent closed itemsets of n-grams, which we will refer to as *frequent patterns of n-grams* in the rest of this paper. We first reproduce the general problem definition found in [10] for clarity, we will then define how it can be applied to text pattern analysis.

Let  $I = \{i_1, i_2, \dots, i_n\}$  be a set of items. An itemset  $X$  is a non-empty subset of  $I$ . Duple  $\langle tid, X \rangle$  is called a transaction if  $tid$  is a transaction identifier and  $X$  is an itemset. An itemset  $X$  is contained in a transaction  $\langle tid, Y \rangle$  if  $X \subset Y$ .

We are interested in finding all the frequent itemsets in the database. An itemset is called frequent if its support is above a given threshold.

In our case, the set  $I$  is the set of all the possible sequences of 3 consecutive words (3-grams) in all the paragraphs of the collection of text. A transaction is a tuple  $\langle par\_id, X \rangle$ , where  $par\_id$  is a paragraph identifier and  $X$  is the set of 3-grams of this paragraph. One frequent itemset is a set of 3-grams that occur together in a minimum number of documents (fixed with a support threshold). Such a set of 3-grams may correspond to an exact repetition in the text or may be a repetition with variations, where only parts of a sentence are exactly repeated but where some “holes” correspond to variations.

Let us consider the set of paragraphs shown in Table 1.

par_id	paragraph
1	I will improve medical aid in our country
2	I will improve security in our country
3	I will improve education in our country

**Table 1: Toy collection with three paragraphs**

Let us consider  $I$ , the set of all 3-grams for these paragraphs:

$I = \{“I will improve”, “will improve medical”, “will improve security”, “will improve education”, “improve medical aid”, “improve security in”, “improve education in”, “medical aid in”, “aid in our”, “security in our”, “education in our”, “in our country”\}$

If we consider a support threshold of 3 paragraphs, then the frequent itemsets are:

$X_1 = \{“I will improve”, “in our country”\}$

$X_2 = \{“I will improve”\}$

$X_3 = \{“in our country”\}$

$X_1$  is an example of a frequent itemset of 3-grams that captures a repetition with slight variations. In the context of a collection of political speeches, we hope that this pattern would invite the user to analyze not only the common parts, but also the differences, which, in this case, are meaningful, i.e. the user may declare: “the speaker is making promises.”

$X_2$  and  $X_3$  are also frequent itemsets but they seem redundant because  $X_1$  carries the same information in one single itemset. We get rid of such smaller itemsets because in case of real documents the number of such sub-patterns could be dramatically high, making their analysis by the user impossible in practice.

According to the following definition,  $X_1$  is a frequent **closed itemset** but  $X_2$  and  $X_3$  are not.

**Definition 3. Closed itemset** [10]: An itemset  $X$  is a closed itemset if there exists no itemset  $X'$  such that:

1.  $X'$  is a proper superset of  $X$ , and
2. Every transaction containing  $X$  also contains  $X'$ .

The following definition is adapted to our work from the definition of closed itemsets.

**Definition 4. Pattern (a closed itemset of 3-grams):** A set of 3-grams  $X$  is a pattern if there exists no set of 3-grams  $X'$  such that:

3.  $X'$  is a proper superset of  $X$ , and
4. Every paragraph containing  $X$  also contains  $X'$ .

### Pattern visualization

In order to help the user interpret patterns made of sets of 3-grams, a tooltip is associated to long patterns in the list. Figure 3 shows an example of the tooltip. The list of 3-grams that compose the pattern is shown in popup window. Each 3-gram is separated by a vertical bar. The first occurrence of the pattern in its context is given as an example to help the user interpret it.

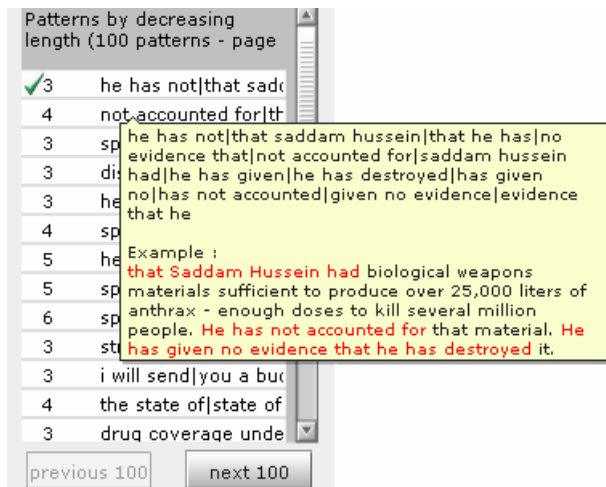


Figure 3: The list of pattern sorted by length, and the tooltip associated to the longest frequent patterns of 3-grams from *The State of the Union* collection.

When a pattern is selected, the paragraphs that contain all the 3-grams of the pattern are colored in the *Documents Overview* panel. The corresponding paragraphs can be displayed in the *Text View* to read the different contexts associated with the pattern. Some paragraphs in the *Text View* may contain only a subset of the n-grams of the pattern; these partial matches are distinguished from exact matches by using different font size. Figure 4 shows three paragraphs that contain the pattern (itemset of 3-grams) shown in Figure 3. A larger font size is used along with coloring to show where an exact match occurs (i.e. all the 3-gram of the selected pattern are contained in the paragraph). Partial matches are also highlighted but they appear with a regular font size.

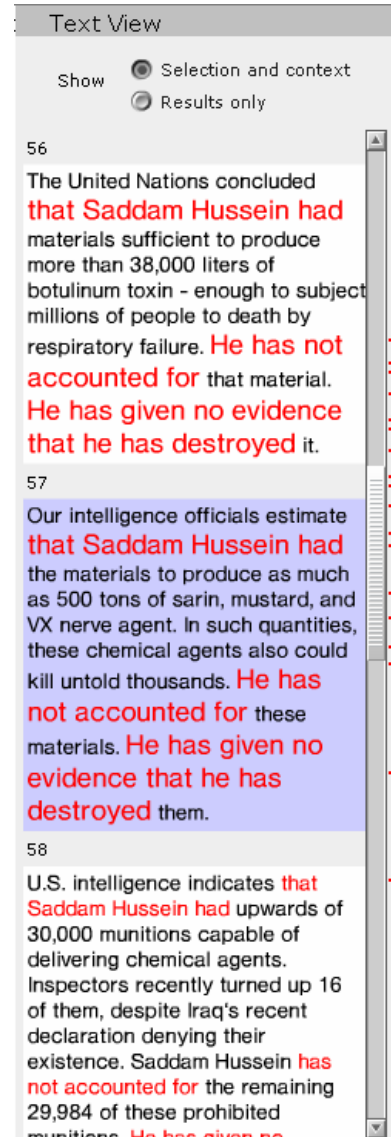


Figure 4: *Text View* panel with two paragraphs that contain an exact match (paragraphs 56 and 57) and one paragraph with only a partial match (paragraph 58).

## 6. ARCHITECTURE

The system was implemented with OpenLaszlo for the client interface and with Ruby and MySQL for the backend part. OpenLaszlo was chosen in order to provide a zero-install access to the user interface and to make FeatureLens accessible from any web browser. Figure 5 shows a sequence diagram of the different parts of the architecture.

The text collections are preprocessed off-line. The frequent words and frequent closed itemsets of 3-grams are computed and stored in a MySQL database together with the text from where they were extracted. OpenLaszlo's visual components are tied to XML files. The XML files may be static or returned by a Web Service over HTTP. The application heavily relies on textual data and full text queries, therefore it needs to:

- 1) Store the text documents in a structured way,
- 2) Have an efficient way to make full-text queries and format the output with text coloring,
- 3) Format the output documents into XML so that OpenLaszlo can use the results.

The Ferret package (a Ruby port of the Lucene tool) was used to build text indices for full-text queries within text documents and the set of frequent patterns. This is very efficient and has a lot of useful features for building text indices (stemming, stop-words filtering) or for querying the index (Boolean queries and sort filters).

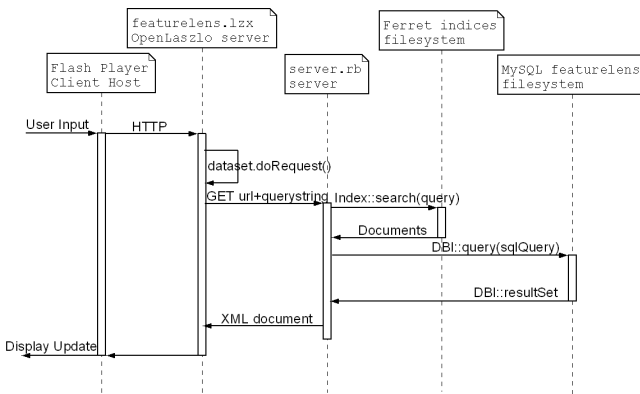


Figure 5: Architecture overview: sequence diagram of the software parts of FeatureLens.

## 7. APPLICATION EXAMPLES

FeatureLens can handle a collection of texts which can be represented as a two-level hierarchy. In a collection of multiple books, the two-level hierarchy can use books and chapters; for a collection made of a single book it can be chapters and sections, in a collection of publication abstracts, year and abstract, etc.

We experimented with different types of text collections. These texts included two books, *The Making of Americans* by Gertrude Stein and *Gamer Theory* by McKenzie Wark, one speech sequence, namely the *State of the Union Addresses* of the U.S. President Bush for the years 2001 through 2007, and abstracts of the research papers published by the University of Maryland

Human-Computer Interaction Lab (HCIL) from 1985 through 2006. Each text collection has its own unique characteristics, and using FeatureLens led to interesting insights for each of them.

The book *The Making of Americans* includes a large number of repetitions. The text is divided into paragraphs and the paragraphs make up the nine sections of the book. Because of the large size of the book, the second level in the hierarchy was chosen to be a unit of five paragraphs instead of one to provide a more compact overview.

The second book, *Gamer Theory*, is a “networked book” created by *The Institute for the Future of the Book*. It is designed to investigate new approaches to writing in a networked environment, when readers and writer are brought together in a conversation about an evolving text. A challenge was set forth to visualize the text, and FeatureLens participated in it<sup>2</sup>. The text consists of nine chapters.

To show FeatureLens’ ability to handle diverse text collection types and to provide interesting but simple examples for our user testing, the *State of the Union Addresses* 2001 through 2007, and the HCIL technical report abstracts from 1984 to 2006 were preprocessed, as well. They were both separated into documents by the publication year.

## 8. PILOT USER EVALUATIONS

FeatureLens was evaluated by performing 2 forms of pilot studies. The tool was used by the literary scholar whose doctoral study deals with the analysis of *The Making of Americans*. In addition a pilot study using *the State of the Union Addresses* was conducted to identify usability problems and see if FeatureLens’ allows users to generate interesting insights about the text collection.

### *The State of the Union Addresses*

The study had eight participants, all of them either had advanced degrees, or were graduate students. Seven had experience in writing computer programs, and five had written software for text analysis. The evaluation consisted of 1) providing the user with background information on FeatureLens and the user study, 2) showing a short demonstration of the interface, 3) allowing the user to explore the text collection with the tool and to comment aloud on the interface and on interesting insights on the text collection. The output from the user study was a list of insights, suggestions for improvement of the tool, and a record of which parts of interface were used during the free exploration.

The exploration had two parts, and it lasted 20 minutes per user unless the user chose to continue further. In the first part, the users were asked to answer two questions:

1. How many times did “terrorist” appear in 2002? The president mentions “the American people” and “terrorist” in the same speeches, did the two terms ever appear in the same paragraph?
2. What was the longest pattern? In which year and paragraphs did it occur? What is the meaning of it?

<sup>2</sup> <http://web.futureofthebook.org/mckenziemark/visualizations>

These questions allowed the users to get acquainted with all the parts of the interface. The users were allowed to ask questions during the exploration, for example, on how to do a particular task or what some part of the interface meant. In the second “free exploration” part, the users were asked to explore the text collection freely, and to comment on their findings.

The goal of the second question was to check if frequent itemsets of 3-grams could be interpreted. The user could find the correct answer via sorting the list of frequent patterns by decreasing size and then, by reading the first element of the list, which is the following set of twelve 3-grams:

$I = \{$ “he has not”, “that saddam hussein”, “that he has”, “no evidence that”, “not accounted for”, “saddam hussein had”, “he has given”, “he has destroyed”, “has given no”, “has not accounted”, “given no evidence”, “evidence that he” $\}$ .

By selecting this pattern from the list, the user could identify three consecutive paragraphs, in the 2003 speech (paragraphs 55, 56 and 57), that contained the twelve 3-grams. Figure 4 shows the details of paragraphs 56 and 57 with the corresponding 3-grams highlighted in red.

All the eight users found correctly and localized the longest pattern in the text collection. Moreover, all of them were able to provide a meaningful interpretation of the pattern, in this case, a repeated expression with variations. This repetition was interpreted by all users as either: an accusation, an attempt to persuade or an attempt to accuse. This example shows that large itemsets of 3-grams can lead to valuable insights, and that they “can” be interpreted with the proposed visualization. Nevertheless it is clear that understanding what a itemsets of 3-grams is is challenging. In our early prototypes which did not provide any example in context (see Figure 3) users were very confused by the long patterns. In our pilot study users were successful in understanding the particular example we selected after receiving a demonstration, but novice users stumbling on such a tool on the internet may still be overwhelmed. While the inclusion of single word patterns alongside the more complex patterns lowered the entry level complexity, video demonstrations and tutorials will be important to help users learn to take advantage of the more advanced pattern mining.

During the free exploration, the users mostly used text queries to find patterns that included specific word of interest, mainly dealing with war, economy and education. Some of the insights came from looking at a single pattern and others required looking at several. Many insights related the appearance of a specific term with another. For example, whenever the President used the phrase “lead the world,” he was referring to environmental topics and not to the “war on terrorism.”. Some other examples include “the President usually means the *U.S. economy* when mentioning *economy*,” “*security* and *congress* occur once together, and it is related to the *Bioshield* project”. These examples illustrate the benefit of visualizing different expressions on the same overview: it helps quickly identify the collocation of different expressions and derive new insights and questions.

Four out of eight users derived some questions from the trends in the distribution of support for particular expressions. Example comments are: “there is a peak in 2005 for the expression *men and women*,” “the term *terror* has a peak in 2002 and *law* has one in

in 2004,” and “before 2002, there is no mention of *economy* whereas there were mentions after the Internet Bubble Crash.” Most users tried to elaborate on these comments by analyzing the context of each expression in order to find an explanation of these trends.

The most common work flow consisted of typing an expression in the search field, then selecting patterns from the returned list. In this case, patterns made out of large itemsets of 3-grams were useful because they provided some contextual information about the searched expression. Surprisingly, most users did not use the sorting by pattern length or frequency during the free exploration, this might be due to the short exploration time (20 minutes) which led users to search for things they had a particular interest in opposed to looking at the default lists of patterns. Finally, the visualization of the trends in the distributions was also used successfully.

Suggestions for improvement included allowing search for 2-grams, exact n-gram search, or using a different color-code for paragraphs where all selected patterns co-occured. One interesting comment concerned the size of frequent patterns of 3-grams. When patterns are sorted by decreasing size, the size corresponds to the number of 3-grams in the pattern and not to the actual number of distinct words. Sorting by the number of distinct words in the pattern might be more appropriate.

### ***The Making of Americans***

We also collected feedback from the early stages of our planned long term case study of the use of FeatureLens with *The Making of Americans*. Our literary expert acted as a design partner during the development of the tool and was eager to test it with her data. There had been dozens of meetings over a period of six months, and the feedback on early interfaces had been incorporated in the version described here. She was particularly interested in frequent patterns with variations and looking forward to sharing her findings with other literary experts. After the prototype reached a satisfactory level of stability, she was able to use it with her data in a free exploration session that lasted two hours. The output from this pilot study was a list of comments and insights about the text.

The first task we encouraged her to tackle was to try to confirm with FeatureLens a finding she had discovered with difficulty with other tools. In this case, her chosen question was about the way the author is referring to the *bottom nature* of her characters (i.e. personality traits). She started by searching for the patterns including the word “kind” then refined the query with a Boolean search (a feature she had requested in early design sessions) “kind NOT men NOT women NOT them”. After scrolling through the resulting list of itemsets of 3-grams, the user easily found and selected *the attacking kind*, *the resisting kind*, *independent dependent kind*, *dependent independent kind* and *engulfing kind of*. These character traits were found in the different sections which describe each character (something the expert already knew); in this way, she was able to confirm that particular personality descriptors could be matched to particular characters.

Afterwards, the expert started to study the way that one particular character is described. The book, *The Making of Americans* is about two families, and in section 1 and 2, the story is centered on the childhood of three members, two brothers and a sister. The

expert identified this part of the text by selecting several house-related terms, which are depicted in Figure 6.

The expert knew the story began with the children and assumed that the words *house* and *governess* would appear in sections 1 and 2. After selecting these terms, the expert noticed that these terms were also occurring together in the first part of section 4. By reading the corresponding paragraphs, she noticed that section 4,

4, which focuses on the sister Martha Hersland, begins with a repetition of the childhood story from sections 1 and 2. The user developed the hypothesis that this reintroduction of the children was probably linked to the story of Martha's failed marriage with her husband *Redfern*. The display of all the occurrences of the word *Redfern* showed that his character was indeed mentioned, just after the repeated section concerning the children.

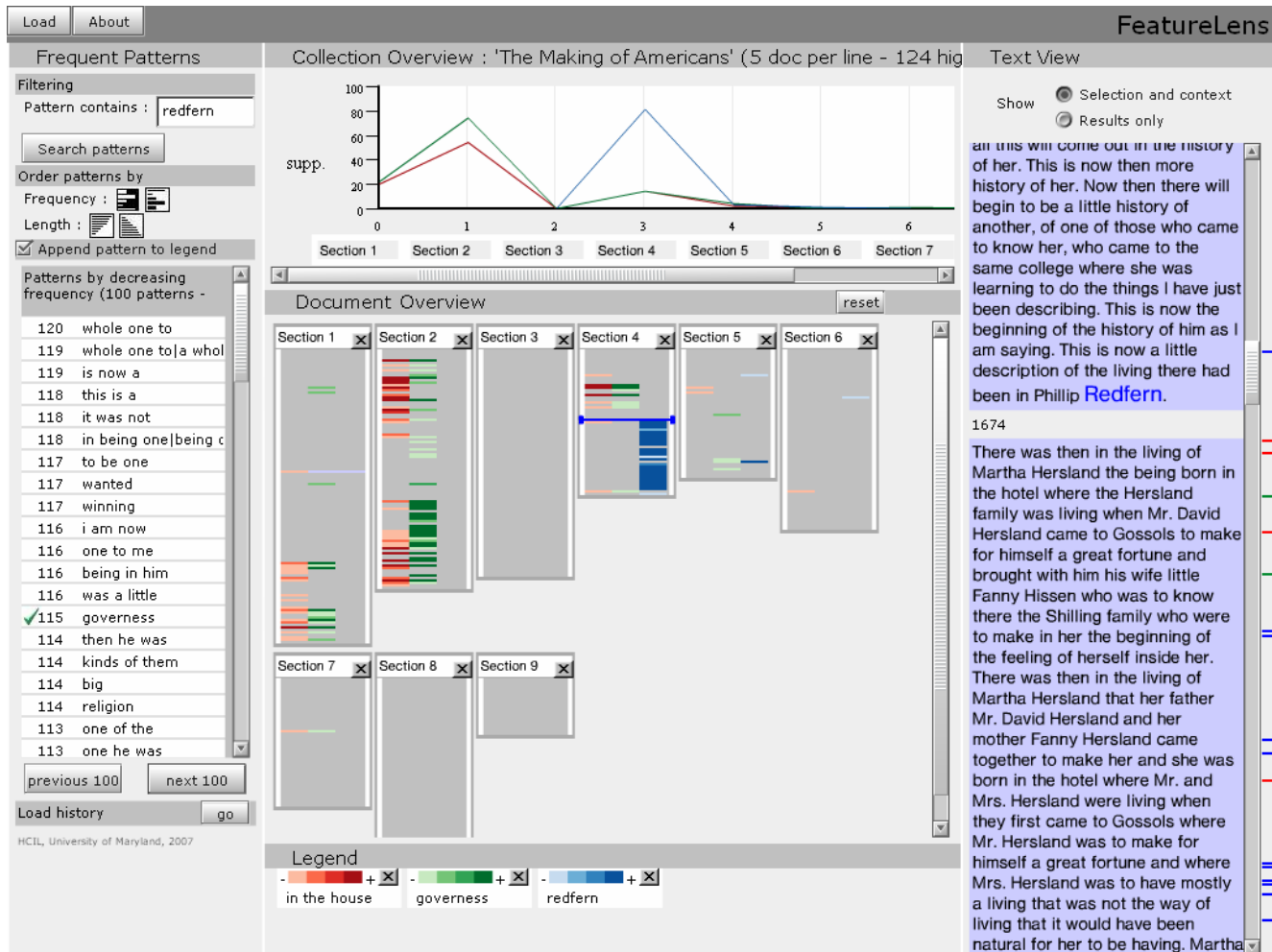


Figure 6: Screen capture of FeatureLens with house-related terms displayed.

Similarly, the expert was able to discover that the author uses the concepts of success and failure when describing marriage in section 6. The words *succeeding*, *failing* and *married* were selected together and the user noticed that the concept of marriage was mentioned in sections 1, 2 and 6, and that it was only associated with the concepts of failure and success in section 6. For the user, this fact supported the idea that the author was describing the marriage factually at the beginning of the book, and then the author introduced a judgment in section 6.

These explorations took place in rapid succession and illustrate how users combine searching, browsing, comparing, and reading in tightly connected ways. The expert was very pleased by what

she had been able to accomplish in a couple of hours, providing some evidence that the proposed system could supports discoveries and lead to useful insight. As for the first group of users, the long frequent itemsets of 3-grams were not examined in the short session, but the shorter itemsets of 3-grams were used extensively in conjunction with text search. A longer case study will shed more light on the potential benefits of the tool.

## 9. CONCLUSION

We described FeatureLens, a system which allows the visual exploration of frequent text patterns in text collections. We applied the concepts of frequent words, frequent expressions and frequent

frequent closed itemsets of n-grams to guide the discovery process. Combined with the interactive visualization, these text mining concepts can help the user to analyze the text, and to create insights and new hypotheses. FeatureLens facilitates the use of itemsets of 3-grams by providing the context of each occurrence. The display of the frequency distribution trends can be used to derive meaningful information. The display of multiple expressions at a time allows studying correlations between patterns.

The user study with *The State of the Union* collection suggests that at first time users use text search as a mean of initial exploration to find patterns of interest, instead of looking at the longest patterns. Being able to display patterns simultaneously was important to make comparisons.

In our future work we will investigate better means of exploration of long patterns and look at more diverse kinds of texts, especially large collections of text where a two level hierarchy may not be sufficient. We will also support the filtering of patterns by their usage trend over time. Metrics can be defined to characterize frequency distributions associated with each pattern and identify that are increasing, decreasing, showing spikes or gaps, etc. Finally, we have focused here on patterns of repetitions, other features can be extracted from the text (e.g. name entities, part of speech patterns) and explored in a similar fashion.

## 10. ACKNOWLEDGMENTS

We would like to thank the volunteers who participated in the user studies for their time and feedback, and Celeste Paul and Matt Kirschenbaum for their feedback and suggestions. Support for this research was provided by the Andrew Mellon Foundation.

## 11. REFERENCES

- [1] Agrawal, R., and R. Srikant, Fast algorithms for mining association rules. In *Proc. 1994 Int. Conf. Very Large Data Bases (VLDB'94)*, 487-499. 1994.
- [2] Church, K.W., and Helfman, J.I., Dotplot: A Program for Exploring Self-Similarity in Millions of Lines of Text and Code, In *Proc. of the 24th Symposium on the Interface, Computing Science and Statistics V24*, 58-67. 1992.
- [3] Eick, S.G. and Steffen, J.L. and Sumner Jr, E.E., Seesoft-A Tool for Visualizing Line Oriented Software Statistics, In *IEEE Transactions on Software Engineering*, Vol 18, No 11, 957-968. 1992.
- [4] Fekete, J. and Dufournaud, N., Compus: visualization and analysis of structured documents for understanding social life in the 16th century. In *Proc. of the Fifth ACM Conference on Digital Libraries*, 47-55. 2000.
- [5] Frank, A. C., Amiri, H., Andersson, S., Genome Deterioration: loss of repeated sequences and accumulation of junk DNA. *Genetica*, Vol. 115, No. 1, 1-12. 2002.
- [6] Kurtz, S & Schleiermacher, C. REPuter: fast computation of maximal repeats in complete genomes. *Bioinformatics* 15, 426-427. 1999.
- [7] G. Lommerse, F. Nossin, L. Voinea, A. Telea, The Visual Code Navigator: An Interactive Toolset for Source Code Investigation. In *Proc. IEEE InfoVis '05*, 24-31. 2005.
- [8] NY Times: The State of the Union in Words. [http://www.nytimes.com/ref/washington/20070123\\_STATEO\\_FUNION.html](http://www.nytimes.com/ref/washington/20070123_STATEO_FUNION.html)
- [9] Paley, W.B. TextArc: Showing Word Frequency and Distribution in Text. *Poster presented at IEEE Symposium on Information Visualization*. 2002.
- [10] J. Pei and J. Han and R. Mao, CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets, *ACM SIGMOD, Workshop on Research Issues in Data Mining and Knowledge Discovery*, 21-30. 2000.
- [11] Plaisant, C. and Rose, J. and Yu, B. and Auvil, L. and Kirschenbaum, M. and Smith, M. and Clement, T. and Lord, G., Exploring Erotics in Emily Dickinson's Correspondence with Text Mining and Visual Interfaces, in *Proc. of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, 141-150. 2006.
- [12] Data to Knowledge (D2K) and Text to knowledge (T2K), NCSA. <http://alg.ncsa.uiuc.edu/do/tools>.
- [13] Thomas, J.J. and Cook, K.A. (eds.), *Illuminating the Path: Research and Development Agenda for Visual Analytics*, IEEE. 2005.
- [14] Veerasamy, A. and Belkin, N. Evaluation of a Tool for Visualization of Information Retrieval Results, in *Proc. of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, 85-92. 1996.
- [15] Wattenberg, M., Arc diagrams: visualizing structure in strings. In *proc IEEE Symposium on Information Visualization 2002*, 110- 116. 2002.
- [16] Wise, J. A. and Thomas, J. J. and Pennock, K. and Lantrip, D. and Pottier, M. and Schur, A. and Crow, V., Visualizing the non-visual: spatial analysis and interaction with information from text documents, In *proc IEEE Symposium on Information Visualization 1995*, 51-58. 1995



Users save “important” messages, ones that announce a new policy at work or the arrival of a friend’s baby. They also save seemingly insignificant messages, ones that suggest we have lunch at 12:30 instead of at noon, ones that involve the logistics of trips long since taken, of meetings long since held [19, 4].

Most tools for handling email archives have focused on the task of finding either specific messages—commercial email clients have search functionalities that let users look for messages by subject or sender—or the “important” emails. Less attention has been paid to the overall patterns of communication that can be gleaned from the daily accumulation of messages over time. PostHistory [17], our first email visualization project, explored temporal patterns of email exchanges. The project revealed that users were quite fascinated by the ability to look back at overall patterns of exchange in their archived messages. This positive feedback prompted us to explore email visualization possibilities even further.

Our hypothesis is that the patterns of communication we build up over time are themselves significant. As email archives grow, they become valuable records of people’s relationships. An increasing amount of our interaction with colleagues, friends, family members, etc. occurs via electronic media such as email.

Whereas face-to-face interactions are rich in sensory detail, online conversations, by contrast, are abstractly sterile. As archives of online interactions “pile up” on a daily basis, users are left with amorphous, homogeneous records of online interactions, little more than white noise. In the same way that we rely on pictures, videos, scrapbooks, and photo albums to remember people and events in real life, we need better ways to capture and depict the email-mediated relationships in which we engage.

In this paper, we describe Themail, a visualization of the contents of an email archive. We also discuss the results from a user study where participants visualized their own email archives. The primary contribution of this paper is the discussion of two main themes that emerged from users’ reactions to Themail: appreciation of the overall picture (“the haystack”) and seeking specific pieces of information (“the needle”). We describe how Themail’s design supported these two themes, and what implications these findings have for the future of email studies in general.

## VISUALIZING EMAIL ARCHIVES

The growth of email archives presents challenges not only to the end user but also to librarians, scholars, historians, forensics experts, and intelligence analysts. It is no surprise then, that research on these collections spans a wide variety of fields: from information management, retrieval and security, to spam detection, social network analysis, and user interface design. More recently, the information

visualization community has also become interested in the idea of exploring email archives and the opportunities they provide for the visual discovery of patterns.

Roughly speaking, most of the work done on email archive visualizations falls into four main categories:

- thread-based visualizations [10, 16]
- social-network visualizations [6, 9, 17]
- temporal visualizations [7, 12, 17]
- contact-based visualizations [11, 15]

As a communication medium, email is inherently suitable for social network analysis (SNA). Moreover, given SNA’s long history of visual exploration [8], it is only natural that researchers came to create email-based social network visualizations with increasing efficiency. Most of this work has been done so that experts can better understand communication patterns in third-party email archives. A good example of this approach is the work being done by team of researchers in Berkeley, which built an entire suite of visualizations for revealing social network patterns in the now public Enron archives [9].

There have also been a few projects that visualize social networks with the end user in mind, an approach that is more in line with the work presented in this paper. Both *Social Network Fragments* [17] and the work done by Fisher and Dourish [6] are ego-centric visualizations of the social network in an individual’s email archive. Both projects were aimed at end users and were tested in a similar manner to Themail.

In addition to visualizing the structure of email networks, researchers have also started to look at different aspects of email chronemics<sup>1</sup> in the hope of finding meaningful patterns of behavior. In [12], researchers uncovered the temporal rhythms of email archives to create context for further email analysis. PostHistory [17], an ego-centric visualization of email archives based on frequency of exchanges, revealed that users could effectively map bursts of email exchange to events in their lives without having to rely on the content of the messages.

Finally, some systems have been developed for contact management – to allow users to keep track of the various people with whom they communicate over email [11, 15].

Themail differs from most email visualizations described here because it relies on the content of messages, instead of on header information, to build a visual display of interactions over time. Visualizations such as [7, 9, 10, 12, 15, 17] depict the patterns and structure of correspondence. Such visualizations are useful for showing the networks of acquaintanceships and the temporal rhythms of interactions, but they do not provide any clues about the topics people discuss or the type of language they use with

---

<sup>1</sup> Chronemics refers to the temporal dimension of communication.





**Figure 2:** Expanded view of Themail showing the sporadic nature of a relationship. “Blank” spaces between columns of words stand for months when no messages were exchanged between the user and the selected email contact.

different members of their social circles. By visualizing the content of messages, Themail creates a more nuanced portrait of the mediated relationship.

Interestingly, outside of the email research area, projects like Conversation Map [13] have explored the validity of using content visualization to make online communities and large-scale conversations more legible.

We developed *Themail* with the working hypothesis that a visualization of email content constituted meaningful portraits of people’s relationships. To test this claim, we needed to let users visualize the relationships encoded in their own email archives. Users’ familiarity with the materials being visualized turned out to be critical in revealing both the successful and the problematic aspects of our content parsing mechanism. We believe that these are valuable insights for researchers working with email content and we discuss the broader implications of our findings to some of the related work happening in email visualization.

## THEMAIL

Themail is a typographic visualization of an individual’s email content over time. The interface shows a series of columns of keywords arranged along a timeline. Keywords are shown in different colors and sizes depending on their frequency and distinctiveness.

The application was designed to help the owners of email archives answer two main questions:

- What sorts of things do I (the owner of the archive) talk about with each of my email contacts?
- How do my email conversations with one person differ from those with other people?

As Themail is designed for the exploration of dyadic relationships, it visualizes one relationship at a time, between the owner of the mailbox, and one of her email contacts.

Themail displays multiple layers of information, each encoding a different content-parsing technique and aesthetic treatment. *Yearly words*—large faint words—

show up in the background, whereas *monthly words*—columns of yellow words—appear in the foreground.

**Yearly words:** reveal the most used terms over an entire year of email exchange.

**Monthly words:** are the most distinctive and frequently used words in email conversations over a month. The selection and font size of words is based not only on frequency but also on how distinctive the word is to a specific relationship against the rest of the archive. For instance, if the owner of the email archive uses the word “environment” a lot with a friend but not with anyone else, the word will appear fairly large when one visualizes this relationship. If, on the other hand, the word “environment” is used a lot with other people in the archive, the word will not be nearly as large in the visualization. The more frequent and distinctive a word is, the bigger it appears in the monthly columns.

## Why monthly and yearly words?

By displaying multiple layers of topical words in different colors and sizes, Themail creates a richly textured portrait of conversations over time. The yearly words in the background, being the most common words used over one year, function as broad brushstrokes in revealing the overall tone of the relationship. For instance, when users in our study visualized their conversational history with family members, several of the yearly words would be terms such as “love,” “hug,” “Thanksgiving,” “family,” etc. Other times, yearly words with a friend would reflect the social nature of the relationship: “dinner,” “tomorrow,” “lunch,” “movie.” Visualizing conversations with a colleague would, many times, generate fairly representative yearly words such as “meeting,” “project,” “deadline.”

Monthly words, on the other hand, revealed a much more detailed portrait of a person’s past email exchanges. Being bound by much shorter periods of time, monthly words successfully depicted the time-based, episodic nature of email conversations. These words clearly depicted the occurrence of events in a relationship. For instance, major events such as an individual’s wedding or preparation for thesis defense were clearly represented in monthly keywords. In several cases a before-and-after-the-fact effect could be seen in the



the person in question has only one email address. In the case above, all of “John Smith’s” email would appear as being from (or to) *john@smith.com*.

Secondly, spam plagues email users. Though many users attempt to filter their mail with spam filters, some spam may remain in their mail archive. Additionally, users who subscribe to many mailing lists will certainly have many mailing list-related messages from people they do not actually know. Because we wanted Themail to focus on *interactive* relationships—meaning, relationships in which the owner of the archive not only receives but also sends out messages—the processing application disregards any email addresses to whom the mailbox owner has not sent at least one email.

### Calculating Topic Words

To generate the words that are at the heart of Themail’s visualization, we use a measure of relative frequency. Salton’s TFIDF algorithm [14], which scores words based on their relative frequency in one document out of a collection, served as a foundation for our process. However, some important differences apply. Namely, in processing monthly and yearly subsets of email, we compared subsets of documents against supersets, rather than one document against a collection.

For each person  $p$ , we compute scores for each word  $w$  in each month  $m$  and year  $y$  that the person exchanged email with the mailbox owner. We compute two scores:

$S_m(w,p,t) = F(w,p,t) * IF(w)$   
score for word  $w$  in all messages to and from person  $p$  in timeslice  $t$ , where  $t$  represents one calendar month.

$S_y(w,p,t) = F(w,p,t)^3 * IF(w)$   
score for word  $w$  in all messages to and from person  $p$  in timeslice  $t$ , where  $t$  represents one calendar year.

These two scoring functions are based on the following measure of frequency:

$F(w,p,t)$  = frequency of word  $w$  in all messages to and from person  $p$  in timeslice  $t$

A word’s inverse frequency is based on its raw count, where  $C(w)$  is the frequency of word  $w$  in all emails in the email archive:

$$IF(w) = \log(1 / C(w))$$

Notice that  $S_m$  and  $S_y$  are the same, except that in  $S_y$ ,  $F(w,p,t)$  is cubed in order to increase its weight in the overall result.

The keywords shown in each month column in the main Themail visualization are selected, sorted, and sized according to  $S_m(w,p,t)$ . The gray words in the background of the visualization are chosen according to  $S_y(w,p,t)$ .

## METHOD

Because we wanted to test Themail “in the wild,” we decided against bringing users into our laboratory. Instead, we distributed the tool via email to participants in the study. Announcements were posted to several mailing lists within universities and research laboratories. No financial compensation was offered for taking the study. Participants were given both the processing and the visualization tools. This approach meant users were able to visualize their own email archives without having to be concerned about the privacy of their data. After having interacted with Themail, users were interviewed about their experience with the tool. These semi-structured interviews lasted 90 minutes each and were recorded for content coding (in the few cases where users were located in different states from where the researchers were, interviews were conducted over email).

### Participants and their archives

Sixteen participants took part in an evaluation of Themail. The subjects ranged in age from 18 to 53; four were female and 12 were male. Participants came from two American universities as well as several technology and telecommunications companies; seven were graduate students and nine were professional researchers. Participants’ email archives ranged in size from 90 MB to more than one GB, with the average size being 456 MB. The time span of these archives ranged from less than one year to over nine years of email activity.<sup>2</sup>

Participants were encouraged to upload multiple mailboxes to Themail – both incoming and outgoing mail. The number of mailboxes uploaded by each participant varied from two to over 55, with the average number being 19.

## RESULTS

### Overview

Overall, participants were quite excited to use Themail to look back at their email archives. When asked, on a scale from 1 to 5 (1 being the least and 5 being the most), how much they enjoyed looking at their email archives on Themail, participants responded, on average 3.9. When asked whether they would like to use the tool again if it were integrated in their email reader, 87% of participants responded yes.

Even though participants were, for the most part, impressed by the quality of the keywords shown in the visualization, they were also quick to point out critical shortcomings in our content parsing mechanism that merit note. We discuss these limitations in the section entitled *Limitations of content parsing in Themail*.

---

<sup>2</sup> Participants in this study were required to have archives that either covered three years of activity (at least) or were at least 100 MB.

Two main interaction modes emerged from the way participants related to Themail. In order to better explain the complimentary nature of these two forms of interaction, we have borrowed terms from the popular expression “looking for a needle in a haystack,” and have called these modes “the haystack” and “the needle.”<sup>3</sup> The former refers to gaining overall understanding and the latter refers to finding specific bits of information. In a sense, this distinction is reminiscent of the division in computer vision between trying to identify specific objects versus understanding scenes (vision for advanced robotics). About 80% of participants used Themail in the *haystack* mode whereas 20% utilized the visualization in the *needle* mode.

In the following sections we describe both the *haystack* and the *needle* interaction approaches and discuss some of the common usage patterns that emerged in each mode.

### The Haystack mode

Users who interacted with Themail in the *haystack* mode, enjoyed using the visualization for the overall picture it presented of their relationships. They usually regarded the visualization as a portrait of their past conversations and frequently drew analogies between Themail and photo albums, in a manner that is reminiscent of our previous email visualization studies [17]. These users often put a premium on being able to see relationships with family members and friends. The more Themail confirmed their expectations—that is, their mental model of what their relationships were like—the more they enjoyed using the tool. This group of participants seemed more interested in overall patterns rather than in picking apart individual words that appeared in the visualization.

In the next subsections, we introduce the main usage patterns to have emerged from this group of users along with case studies of participants whose comments clearly illustrate the usage patterns being discussed.

#### Data as Portrait

Most haystack users appreciated having expectations of their relationships confirmed by the visualization while still being able to drill deeper and discover patterns they were not aware of. Several people enjoyed most looking at their families and friends in the visualization and comparing what they saw on screen with their impressions of these relationships.

*The best "portrait" was for the mail with my mother... There have been all sorts of emotional things happening in the past few months (her mother/my grandmother passed away, she had surgery, etc.) and all of that comes through dramatically in the visualization.*

*If you look at the ten first words of each monthly column, for instance, it's like you are following someone's life story.*

#### Case Study: Ann<sup>4</sup>

Ann is a graduate student in an American university. She is 26 years old and has recently gotten married. Her extended family lives in the south of the United States and she lives with her husband in New England. For Ann, one of the most exciting aspects of Themail was seeing all the correspondence that preceded her wedding:

*It was funny going back both with [my husband] and my parents, there were these few months before our wedding... it's all about the wedding! There are all these words like "invitations," "tables," "drinks," "guests" names. It's got all these words that are totally related to the wedding plans. It was all in October and November [user gestures a peak] and then the words completely changed after that. And the same happened with my friends that were bridesmaids. There are these few months where you can see that the words were related to our wedding theme but then, the month after the conversation it all switched back to normal. Yeah, it was like the before and after. You could definitely see the event.*

Ann thought it was important that Themail allowed her to look back at her relationships with loved ones, friends, and family. Even though she exchanges more emails with her coworkers on a daily basis, it was the personal facet of her email archive that she felt was the most exciting to explore.

*Especially for my family, it was really exciting to see all the words and the things that we talk about for no reason other than to just reminisce; it was like looking through a photo album or something. For instance, I would never go back and search for the wedding planning emails, but it was fun to look at that! It's almost like this serves a different kind of purpose from regular email readers... It's more at a personal level... It's emotional, it's about reflecting and remembering.*

After looking at her correspondence with family members, Ann remarked that some “portraits” read very differently from others. With her brother, for instance, the themes ranged from talking about his kids to him asking Ann for help with his computer. With her grandmother, however, the words that came up on Themail referred to religious holidays and themes.

*[Grandmother] was interesting... I don't even remember, if you asked me, what kinds of emails I've exchanged with my grandmother; we don't write email all the time, and a lot of times the news flow through my Mom. But I felt like her Themail visualization really characterized her. It was probably because she was a whole lot different than anyone*

---

<sup>3</sup> We thank Martin Wattenberg for pointing out the “needle & haystack” metaphor in the context of our email research.

---

<sup>4</sup> All identifying information, including names of participants, their family members, acquaintances, and, where necessary, topical keywords, have been changed to preserve anonymity.



*My internship student: this is very interesting (especially in the expanded view) – it shows the period for arranging the interview, day to day work emails and recent contact re-establishing the link.*

### **New Perspectives on Relationships**

The sheer collection of words exchanged with a person made the texture of different relationships quickly obvious to users. By looking at these compilations of words, users were able to gain a new perspective on their relationships:

*This is the one I got a little chuckle out of... this is the [ethnic dance] mailing list; this is a group of us who help manage a dance club at [our university]. And the thing that sort of stood out to me here was the fact that, just about every one of these columns has the word 'please' which is a reflection of everyone begging each other to do something! It's like, 'you guys, please do this, please do that...' and so, I thought it was really funny that this was sort of a predominant word that we're all just begging each other to do stuff! That's really what this is about. [Figure 1]*

*This one reminded me of the fact that I was a slacker for the first couple of years [of my PhD program] and stuff like that... [Interviewer asks: how did the visualization remind you of that?] Well, because the name of our baseball team appears here! I'm talking more about baseball with [my advisor] than I am about work. I should probably have been working a little harder back then.*

### **The Needle mode**

About 20% of participants in the study were more interested in finding specific bits of information rather than focusing on the overall patterns of the visualization (*haystack* users). Participants in this category displayed little interest in looking at the visualization of their family members, being more concerned with visualizing work-related relationships:

*Instead of seeing my daily conversations with my wife I would rather be able to see that in 2002 I wrote a paper with Bob and this is what we talked about at that point and we haven't talked about any of that anymore, so on and so forth. That is why I don't think there are any big surprises [in Themail] because this is what I know without even having to look at a visualization.*

The last sentence in the quote above illustrates the main difference between *haystack* and *needle* users: the former take pleasure in seeing a tangible depiction of what they already know whereas the latter are more interested in discovering what they did not know or remember.

At first, *needle* users seemed a bit underwhelmed by Themail, however, when prompted to talk in detail about what the visualization showed of their relationship with top email contacts, most of these users became surprised by the richness of detail in the Themail keywords:

*I'm not sure where the word 'femur' came from; why would I be talking with my dad about 'femur'? [The user clicks on the word and reads the messages that contain 'femur'] Ah...my grandmother got hurt; that's right. This is her name here [pointing at the visualization].*

*I saw the word "horse" in the collection of correspondence with a family member and assumed that the email would be about the horses my brother has on his small farm in Minnesota. As it turns out, the email was one from my daughter (using my email account) describing the horse riding lessons she had just begun. Nice turn of events, as her email was written in the voice of a small child (~ 10 years old).*

*I clicked on the work "decision" in an email from a friend of mine who worked with me on a local school technology planning committee. I was curious about what we might have had to "decide" about, and sure enough, it was an email about the MAC vs. Windows platform "decision" for the school. This brought back memories of many long, and quite heated discussions on the topic among parents, teachers and members of the committee.*

Such discovery episodes demonstrate that, even though Themail was not designed for querying data, its abundance of easy-to-get-to information let users find interesting bits of data quite effortlessly. Whenever users saw words that seemed out of place given the context of their relationships, they would invariably click on them to find out whether the system had made a mistake. Almost always, like in the above quote about the word "femur," users would be reminded of events they had forgotten. In fact, the ability to select keywords and see the email messages that caused those words to appear in the visualization was, more than any other aspect of Themail, the single feature that succeeded in strengthening users' trust in the visualization.

### **Limitations of content parsing in Themail**

Other than the messages disregarded by the processing tool<sup>5</sup>, the content analysis algorithm in Themail treats every message in the same way. This means that there are no messages with special weights or attributes in the keyword scoring mechanism. As it became clear from the user study, not all messages are created equal, and our egalitarian approach presents some serious limitations in terms of keyword output.

One of the main problems that participants identified on their Themail visualizations was the inadvertently high weight given to topical words in forwarded messages. For instance, sometimes the unique words in jokes that had

---

<sup>5</sup> Messages originating from people to whom the owner of the archive has never sent a message are automatically deleted from the dataset processed by the Themail content parsing tool.

been forwarded to the owner of the email ended up having too much weight, becoming the focus of the visualization. Whenever this was the case, participants remarked that Themail did a poor job of representing their email conversations. This phenomenon was not limited to forwarded messages, having also happened with sent-out announcements and pieces of code inserted in the body of email messages. In one extreme case, Themail displayed the relationship between an administrative assistant and the owner of the email as being studded with some of the most academic and highbrow words in that person's email archive. In fact, the administrative assistant had the task of sending out announcements for every thesis defense in that university department. The focal words in the visualization came straight from students' dissertation abstracts rather than having been produced by email exchanges between the owner of the email and the administrative assistant.

Unrepresentative keywords also came out of people's email signatures. Methods for removing signatures from emails include identifying email addresses and other contact information, as well as a large amount of non-alphanumeric characters, such as punctuation [2]. Themail accomplished some of this by ignoring URLs, email addresses and numeric strings. Signature content persisted in Themail most often when the signature (1) contained quotations, e.g. from famous people or song lyrics, (2) changed over time, or (3) varied according to multiple addresses the person held. In such cases, Themail would treat the words in the signature as content.

The second major limitation in our content parsing mechanism is granularity. Themail has no notion of expressions; it only knows individual words. This approach imposes clear limits to the output depicted in Themail as the --tone and texture of messages is more fully expressed by phrases rather than by separate words. A desirable next step for the work presented here would be to have it analyze phrases.

## **Implications for HCI**

### *Methodology*

Information visualization is generally used for understanding unfamiliar, complex data spaces. In effectively displaying overviews of large datasets, visualizations quickly reveal unknown patterns in the data. In our study of Themail, however, we distributed the tool to users who were *already familiar* with the datasets being visualized. Participants had some idea of what to expect in the visualization: they anticipated they would see the names of the people with whom they emailed the most, different kinds of words reflecting different kinds of relationships, etc. Even though building a tool to visualize familiar data is not the conventional course of action in information visualization, we feel that it is a valuable approach in email research and one that can lead to important advances in the field.

By testing Themail with users who were familiar with the datasets being visualized, we were able to learn about important limitations of our content parsing algorithm. Because users knew their email archives well, they were able to quickly point out some key problems that would have taken us much longer to discern. By distributing Themail to users, we made sure that it was tested in a natural setting—as opposed to having been tested at a laboratory—and that users did not have to worry about the privacy of their data.

We believe that this approach can be of value to other work being done in this area as well. For example, email visualizations that are built for the discovery of patterns by third-party experts, can also take advantage of insights from users who are familiar with the email archives being visualized. If, for instance, a visualization designer tests her expert system with owners of email archives—not her target audience but, rather, a “testing unit”—she might quickly learn about the kinds of patterns her tool displays and any potentially problematic artifacts the system generates. Most current studies of email visualization would be greatly complemented by this kind of approach.

### *Content Analysis*

Text analysis is a vast and complex area of research. Our contribution is not a particular algorithm but rather a preliminary understanding of what is distinctive in the analysis of email content.

Results from the Themail user study made it clear that content parsing of email exchanges is inherently different from other kinds of text analysis. Email messages are not created equal and their differences need to be taken into account when exploring content. For instance, we may want to consider the content of forwarded messages and announcements differently from the content of messages exchanged between friends. Our user study revealed that participants were highly aware of and sensitive to these differences.

Some of these dimensions can be extracted from traffic patterns, chronemics, and the symmetry of exchanges. If such patterns were to be integrated with email content analysis, we could likely generate much more representative portrayals of email relationships. For instance, it would be useful to utilize traffic patterns to decide when an email exchange should be considered a conversation as opposed to a broadcast. If we can generate conversational models that take these structural elements into account, we will be in a better position to meaningfully analyze the content of email exchanges.

## **CONCLUSION**

We have presented Themail, an original approach to email archive visualization that uses content to portray individual

relationships. Our user study revealed two main interaction modes with the visualization, exploration of “big picture” trends and themes (“haystack”) and more detail-oriented exploration (“needle”). The overwhelming majority of participants utilized the system in the “haystack” mode and were especially fond of looking at their relationships with family members and loved ones. These users often remarked on the photo-album quality of Themail and said that they would like to share the visualization with others. Users in the “needle” mode were more interested in finding specific bits of information in the visualization; they were especially interested in being able to identify information that was work related.

By displaying large collections of keywords that reflected the evolution of relationships over time, Themail placed users’ past email exchanges within a meaningful context. Given that email is a habitat [5] and that an increasing amount of our daily interactions with others occur via email, the supplementary contextual cues presented in Themail can greatly improve users’ utilization of email archives. We do not, however, expect users to utilize applications such as Themail on a daily basis. Given the analogy with photographs, it seems more likely that users might engage in sporadic explorations of the display to reminisce.

We relied on participants’ familiarity with the data for effectively learning about some of the shortcomings of our content analysis algorithm. We propose that recognizing the importance of personal identification with the data is a key contribution to email content studies in particular and email research in general.

#### ACKNOWLEDGMENTS

We are indebted to Shreyes Seshasai and Daniel Martines for their help in implementing Themail. We would also like to thank the participants in our study for allowing us to share their thoughts on Themail.

#### REFERENCES

1. Bellotti, V., Ducheneaut, N., Howard, M., & Smith, I. (2003). *Taking Email to Task: The Design and Evaluation of a Task Management Centered Email Tool*. In Proc CHI.
2. Carvalho, V. & Cohen, W. (2004) *Learning to Extract Signature and Reply Lines from Email*. Conference on Email and Anti-Spam.
3. Csikszentmihalyi, M., and Rochberg-Halton, E. 1981. *The Meaning of Things*. Cambridge University Press, Cambridge, UK.

4. Dabbish, L., Kraut, R., Fussell, S. & Kiesler, S. (2005) *Understanding email use: predicting action on a message*. In SIGCHI, ACM Press.
5. Ducheneaut, N., & Bellotti, V. (2001). *Email as habitat: an exploration of embedded personal information management*. Interactions, 8(5), pp. 30-38.
6. Fisher, D., & Dourish, P. (2004). *Social and Temporal Structures in Everyday Collaboration*. In Proc. CHI.
7. Frau, S., Roberts, J., & Boukhelifa, N. (2005). *Dynamic Coordinated Email Visualization*. In WSCG.
8. Freeman, L. (2000). *Visualizing Social Networks*. Journal of Social Structure, 1 (1).
9. Heer, J. *Exploring Enron: Visual Data Mining of E-mail*. Available online at <http://jheer.org/enron/>
10. Kerr, B. (2003) *Thread Arcs: An Email Thread Visualization*. IBM Research Report.
11. Nardi, B., Whittaker, S., Isaacs, E., Creech, M., Johnson, J., & Hainsworth, J. (2002). *ContactMap: Integrating Communication and Information Through Visualizing Personal Social Networks*. Communications of the ACM.
12. Perer, A., Shneiderman, B., & Oard, D. (2005) *Using Rhythms of Relationships to Understand Email Archives*. In Review.
13. Sack, W. (2000) *Conversation Map: An Interface for Very-Large-Scale Conversations*. Journal of Management Information Systems, Vol 17, No. 3.
14. Salton, G. (1989) *Automatic Text Processing - The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley.
15. Sudarsky, S., & Hjelsvold, R. (2002) *Visualizing Electronic Email*. In International Conference on Information Visualization.
16. Venolia, G. & Neustaedter, C. (2003) *Understanding sequence and reply relationships within email conversations: a mixed-model visualization*. In SIGCHI.
17. Viégas, F., boyd, d., Nguyen, D., Potter, J. & Donath, J. (2004) *Digital Artifacts for Remembering and Storytelling: PostHistory and Social Network Fragments*. In HICSS-37.
18. Whittaker, S. & Hirschberg, J. (2001) *The character, value, and management of personal paper archives*. In ACM TOCHI.
19. Whittaker, S. & C. Sidner Year (1996) *Email overload: Exploring personal information management of email*. In SIGCHI.



# User-directed Sentiment Analysis: Visualizing the Affective Content of Documents

**Michelle L. Gregory**

PNNL  
902 Battelle Blvd.  
Richland Wa. 99354  
michelle.gregory@pnl.gov

**Nancy Chinchor**

Consultant  
chinchor@earthlink.net

**Paul Whitney**

PNNL  
902 Battelle Blvd.  
Richland Wa. 99354  
paul.whitney@pnl.gov

**Richard Carter**

PNNL  
902 Battelle Blvd.  
Richland Wa. 99354  
richard.carter@pnl.gov

**Elizabeth Hetzler**

PNNL  
902 Battelle Blvd.  
Richland Wa. 99354  
beth.hetzler@pnl.gov

**Alan Turner**

PNNL  
902 Battelle Blvd.  
Richland Wa. 99354  
alan.turner@pnl.gov

## Abstract

Recent advances in text analysis have led to finer-grained semantic analysis, including *automatic sentiment analysis*—the task of measuring documents, or chunks of text, based on emotive categories, such as *positive* or *negative*. However, considerably less progress has been made on efficient ways of exploring these measurements. This paper discusses approaches for visualizing the affective content of documents and describes an interactive capability for exploring emotion in a large document collection.

## 1 Introduction

Recent advances in text analysis have led to finer-grained semantic classification, which enables the automatic exploration of subtle areas of meaning. One area that has received a lot of attention is *automatic sentiment analysis*—the task of classifying documents, or chunks of text, into emotive categories, such as *positive* or *negative*. Sentiment analysis is generally used for tracking people’s attitudes about particular individuals or items. For example, corporations use sentiment analysis to determine employee attitude and customer satisfaction with their products. Given the plethora of data in digital form, the ability to accurately and efficiently measure the emotional content of documents is paramount.

The focus of much of the automatic sentiment analysis research is on identifying the *affect bearing* words (words with emotional content) and on measurement approaches for sentiment (Turney & Littman, 2003; Pang & Lee, 2004; Wilson et al., 2005). While identifying related

content is an essential component for automatic sentiment analysis, it only provides half the story. A useful area of research that has received much less attention is how these measurements might be presented to the users for exploration and added value.

This paper discusses approaches for visualizing affect and describes an interactive capability for exploring emotion in a large document collection. In Section 2 we review current approaches to identifying the affective content of documents, as well as possible ways of visualizing it. In Section 3 we describe our approach: The combination of a lexical scoring method to determine the affective content of documents and a visual analytics tool for visualizing it. We provide a detailed case study in Section 4, followed by a discussion of possible evaluations.

## 2 Background

At the AAAI Symposium on Attitude and Affect held at Stanford in 2004 (Qu et al., 2005), it was clear that the lexical approach to capturing affect was adequate for broad brush results, but there were no production quality visualizations for presenting those results analytically. Thus, we began exploring methods and tools for the visualization of lexically-based approaches for measuring affect which could facilitate the exploration of affect within a text collection.

### 2.1 Affect Extraction

Following the general methodology of informational retrieval, there are two pre-dominant methods for identifying sentiment in text: Text classification models and lexical approaches. Classification models require that a set of documents are hand labeled for affect, and a system is

trained on the feature vectors associated with labels. New text is automatically classified by comparing the feature vectors with the training set. (Pang & Lee, 2004; Aue & Gamon, 2005). This methodology generally requires a large amount of training data and is domain dependent.

In the lexical approach, documents (Turney & Littman, 2003), phrases (see Wilson et al., 2005), or sentences (Weibe & Riloff, 2005) are categorized as *positive* or *negative*, for example, based on the number of words in them that match a lexicon of sentiment bearing terms. Major drawbacks of this approach include the contextual variability of sentiment (what is *positive* in one domain may not be in another) and incomplete coverage of the lexicon. This latter drawback is often circumvented by employing *bootstrapping* (Turney & Littman, 2003; Weibe & Riloff, 2005) which allows one to create a larger lexicon from a small number of seed words, and potentially one specific to a particular domain.

## 2.2 Affect Visualization

The uses of automatic sentiment classification are clear (public opinion, customer reviews, product analysis, etc.). However, there has not been a great deal of research into ways of visualizing affective content in ways that might aid data exploration and the analytic process.

There are a number of visualizations designed to reveal the emotional content of text, in particular, text that is thought to be highly emotively charged such as conversational transcripts and chat room transcripts (see DiMicco et al., 2002; Tat & Carpendale, 2002; Lieberman et al., 2004; Wang et al., 2004, for example). Aside from using color and emoticons to explore individual documents (Liu et al., 2003) or email inboxes (Mandic & Kerne, 2004), there are very few visualizations suitable for exploring the affect of large collections of text. One exception is the work of Liu et al. (2005) in which they provide a visualization tool to compare reviews of products, using a bar graph metaphor. Their system automatically extracts product features (with associated affect) through parsing and pos tagging, having to handle exceptional cases individually. Their Opinion Observer is a powerful tool designed for a single purpose: comparing customer reviews.

In this paper, we introduce a visual analytic tool designed to explore the emotional content of large collections of open domain documents. The tools described here work with document collections of all sizes, structures (html, xml, .doc,

email, etc), sources (private collections, web, etc.), and types of document collections. The visualization tool is a mature tool that supports the analytical process by enabling users to explore the thematic content of the collection, use natural language to query the collection, make groups, view documents by time, etc. The ability to explore the emotional content of an entire collection of documents not only enables users to compare the range of affect in documents within the collection, but also allows them to relate affect to other dimensions in the collection, such as major topics and themes, time, and source.

## 3 The Approach

Our methodology combines a traditional lexical approach to scoring documents for affect with a mature visualization tool. We first automatically identify affect by comparing each document against a lexicon of affect-bearing words and obtain an affect score for each document. We provide a number of visual metaphors to represent the affect in the collection and a number of tools that can be used to interactively explore the affective content of the data.

### 3.1 Lexicon and Measurement

We use a lexicon of affect-bearing words to identify the distribution of affect in the documents. Our lexicon authoring system allows affect-bearing terms, and their associated strengths, to be bulk loaded, declared manually, or algorithmically suggested. In this paper, we use a lexicon derived from the General Inquirer (GI) and supplemented with lexical items derived from a semi-supervised bootstrapping task. The GI tool is a computer-assisted approach for content analyses of textual data (Stone, 1977). It includes an extensive lexicon of over 11,000 hand-coded word stems and 182 categories.

We used this lexicon, specifically the *positive* and *negative* axes, to create a larger lexicon by bootstrapping. Lexical bootstrapping is a method used to help expand dictionaries of semantic categories (Riloff & Jones, 1999) in the context of a document set of interest. The approach we have adopted begins with a lexicon of affect bearing words (POS and NEG) and a corpus. Each document in the corpus receives an affect score by counting the number of words from the seed lexicon that occur in the document; a separate score is given for each affect axis. Words in the corpus are scored for affect potential by comparing their distribution (using an L1 Distri-

bution metric) of occurrence over the set of documents to the distribution of affect bearing words. Words that compare favorably with affect are hypothesized as affect bearing words. Results are then manually culled to determine if in fact they should be included in the lexicon.

Here we report on results using a lexicon built from 8 affect categories, comprising 4 concept pairs:

- Positive ( $n=2236$ )-Negative ( $n=2708$ )
- Virtue ( $n=638$ )-Vice ( $n=649$ )
- Pleasure ( $n=151$ )-Pain ( $n=220$ )
- Power Cooperative ( $n=103$ )-Power Conflict ( $n=194$ )

Each document in the collection is compared against all 8 affect categories and receives a score for each. Scores are based on the summation of each affect axis in the document, normalized by the number of words in the documents. This provides an overall proportion of *positive* words, for example, per document. Scores can also be calculated as the summation of each axis, normalized by the total number of affect words for all axes. This allows one to quickly estimate the balance of affect in the documents. For example, using this measurement, one could see that a particular document contains as many *positive* as *negative* terms, or if it is heavily skewed towards one or the other.

While the results reported here are based on a predefined lexicon, our system does include a *Lexicon Editor* in which a user can manually enter their own lexicon or add strengths to lexical items. Included in the editor is a *Lexicon Bootstrapping Utility* which the user can use to help create a specialized lexicon of their own. This utility runs as described above. Note that while we enable the capability of strength, we have not experimented with that variable here. All words for all axes have a default strength of .5.

## 3.2 Visualization

To visualize the affective content of a collection of documents, we combined a variety of visual metaphors with a tool designed for visual analytics of documents, IN-SPIRE.

### 3.2.1 The IN-SPIRE System

IN-SPIRE (Hetzler and Turner, 2004) is a visual analytics tool designed to facilitate rapid understanding of large textual corpora. IN-SPIRE generates a compiled document set from *mathematical signatures* for each document in a set.

Document signatures are clustered according to common themes to enable information exploration and visualizations. Information is presented to the user using several *visual metaphors* to expose different facets of the textual data. The central visual metaphor is a **Galaxy view** of the corpus that allows users to intuitively interact with thousands of documents, examining them by theme (see Figure 4, below). IN-SPIRE leverages the use of context vectors such as LSA (Deerwester et al., 1990) for document clustering and projection. Additional analytic tools allow exploration of temporal trends, thematic distribution by source or other metadata, and query relationships and overlaps. IN-SPIRE was recently enhanced to support visual analysis of sentiment.

### 3.2.2 Visual Metaphors

In selecting metaphors to represent the affect scores of documents, we started by identifying the kinds of questions that users would want to explore. Consider, as a guiding example, a set of customer reviews for several commercial products (Hu & Liu, 2004). A user reviewing this data might be interested in a number of questions, such as:

- What is the range of affect overall?
- Which products are viewed most positively? Most negatively?
- What is the range of affect for a particular product?
- How does the affect in the reviews deviate from the norm? Which are more negative or positive than would be expected from the averages?
- How does the feedback of one product compare to that of another?
- Can we isolate the affect as it pertains to different features of the products?

In selecting a base metaphor for affect, we wanted to be able to address these kinds of questions. We wanted a metaphor that would support viewing affect axes individually as well as in pairs. In addition to representing the most common axes, negative and positive, we wanted to provide more flexibility by incorporating the ability to portray multiple pairs because we suspect that additional axes will help the user explore nuances of emotion in the data. For our current metaphor, we drew inspiration from the Rose plot used by Florence Nightingale (Wainer, 1997). This metaphor is appealing in that it is easily interpreted, that larger scores draw more

attention, and that measures are shown in consistent relative location, making it easier to compare measures across document groups. We use a modified version of this metaphor in which each axis is represented individually but is also paired with its opposite to aid in direct comparisons. To this end, we vary the spacing between the rose petals to reinforce the pairing. We also use color; each pair has a common hue, with the more positive of the pair shown in a lighter shade and the more negative one in a darker shade (see Figure 1).

To address how much the range of affect varies across a set of documents, we adapted the concept of a box plot to the rose petal. For each axis, we show the median and quartile values as shown in the figure below. The dark line indicates the median value and the color band portrays the quartiles. In the plot in Figure 1, for example, the scores vary quite a bit.

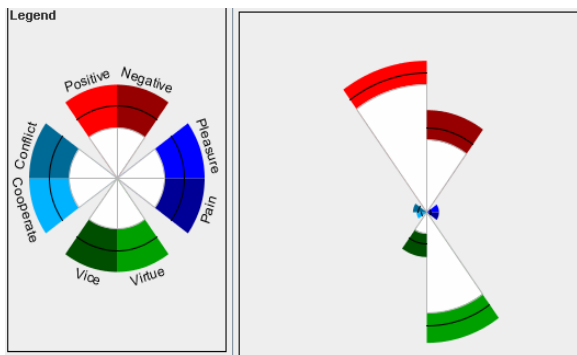


Figure 1. Rose plot adapted to show median and quartile variation.

Another variation we made on the base metaphor was to address a more subtle set of questions. It may happen that the affect scores within a dataset are largely driven by document membership in particular groups. For example, in our customer data, it may be that all documents about Product A are relatively positive while those about Product B are relatively negative. A user wanting to understand customer complaints may have a subtle need. It is not sufficient to just look at the most negative documents in the dataset, because none of the Product A documents may pass this threshold. What may also help is to look at all documents that are more negative than one would expect, given the product they discuss. To carry out this calculation, we use a statistical technique to calculate the Main (or expected) affect value for each group and the Residual (or deviation) affect value for each document with respect to its group (Scheffe, 1999).

To convey the Residual concept, we needed a representation of deviation from expected value. We also wanted this portrayal to be similar to the base metaphor. We use a unit circle to portray the expected value and show deviation by drawing the appropriate rose petals either outside (larger than expected) or inside (smaller than expected) the unit circle, with the color amount showing the amount of deviation from expected. In the figures below, the dotted circle represents expected value. The glyph on the left shows a cluster with scores slightly higher than expected for Positive and for Cooperation affect. The glyph on the right shows a cluster with scores slightly higher than expected for the Negative and Vice affect axes (Figure 2).

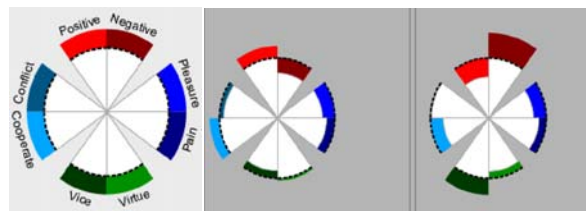


Figure 2. Rose plot adapted to show deviation from expected values.

### 3.2.3 Visual Interaction

IN-SPIRE includes a variety of analytic tools that allow exploration of temporal trends, thematic distribution by source or other metadata, and query relationships and overlaps. We have incorporated several interaction capabilities for further exploration of the affect. Our analysis system allows users to group documents in numerous ways, such as by query results, by metadata (such as the product), by time frame, and by similarity in themes. A user can select one or more of these groups and see a summary of affect and its variation in those groups. In addition, the group members are clustered by their affect scores and glyphs of the residual, or variation from expected value, are shown for each of these sub-group clusters.

Below each rose we display a small histogram showing the number of documents represented by that glyph (see Figure 3). These allow comparison of affect to cluster or group size. For example, we find that extreme affect scores are typically found in the smaller clusters, while larger ones often show more mid-range scores. As the user selects document groups or clusters, we show the proportion of documents selected.



Figure 3. Clusters by affect score, with one rose plot per cluster.

The interaction may also be driven from the affect size. If a given clustering of affect characteristics is selected, the user can see the themes they represent, how they correlate to metadata, or the time distribution. We illustrate how the affect visualization and interaction fit into a larger analysis with a brief case study.

#### 4 Case study

The IN-SPIRE visualization tool is a non-data specific tool, designed to explore large amounts of textual data for a variety of genres and document types (doc, xml, etc). Many users of the system have their own data sets they wish to explore (company internal documents), or data can be harvested directly from the web, either in a single web harvest, or dynamically. The case study and dataset presented here is intended as an example only, it does not represent the full range of exploration capabilities of the affective content of datasets.

We explore a set of customer reviews, comprising a collection of Amazon reviews for five products (Hu & Liu, 2004). While a customer may not want to explore reviews for 5 different product types at once, the dataset is realistic in that a web harvest of one review site will contain reviews of multiple products. This allows us to demonstrate how the tool enables users to focus on the data and comparisons that they are interested in exploring. The 5 products in this dataset are:

- Canon G3; digital camera
- Nikon coolpix 4300; digital camera
- Nokia 6610; cell phone
- Creative Labs Nomad Jukebox Zen Xtra 40GB; mp3 player
- Apex AD2600 Progressive-scan DVD player

We begin by clustering the reviews, based on overall thematic content. The labels are automatically generated and indicate some of the stronger theme combinations in this dataset. These clusters are driven largely by product vocabulary. The two cameras cluster in the lower portion; the Zen shows up in the upper right clusters, with the phone in the middle and the Apex DVD player in the upper left and upper middle. In this image, the pink dots are the Apex DVD reviews.

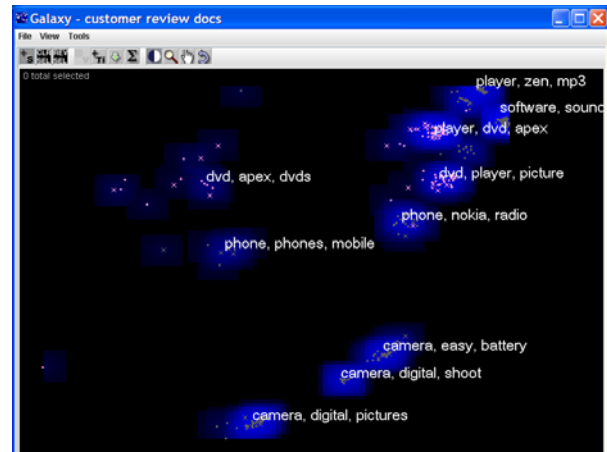


Figure 4. Thematic clustering of product review

The affect measurements on these documents generate five clusters in our system, each of which is summarized with a rose plot showing affect variation. This gives us information on the range and distribution of affect overall in this data. We can select one of these plots, either to review the documents or to interact further. Selection is indicated with a green border, as shown in the upper middle plot of Figure 5.

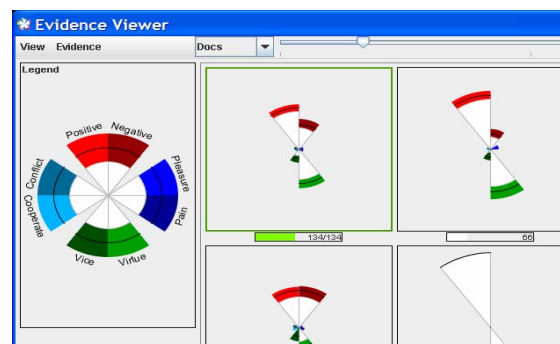


Figure 5. Clusters by affect, with one cluster glyph selected.

The selected documents are relatively positive; they have higher scores in the Positive and Virtue axes and lower scores in the Negative axis. We may want to see how the documents in this

affect cluster distribute over the five products. This question is answered by the correlation tool, shown in Figure 6; the positive affect cluster contains more reviews on the Zen MP3 player than any of the other products.

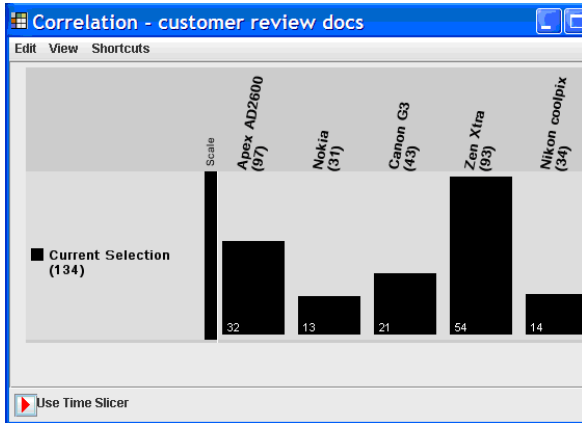


Figure 6. Products represented in one of the positive affect clusters.

Alternatively we could get a summary of affect per product. Figure 7 shows the affect for the Apex DVD player and the Nokia cell phone. While both are positive, the Apex has stronger negative ratings than the Nokia.

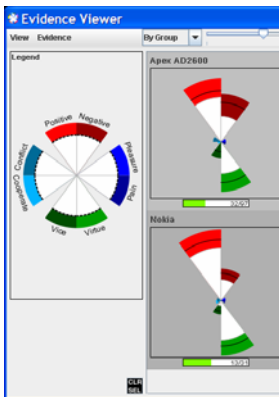


Figure 7. Comparison of Affect Scores of Nokia to Apex

More detail is apparent by looking at the clusters within one or more groups and examining the deviations. Figure 8 shows the sub-clusters within the Apex group. We include the summary for the group as a whole (directly beneath the Apex label), and then show the four sub-clusters by illustrating how they deviate from expected value. We see that two of these tend to be more positive than expected and two are more negative than expected.

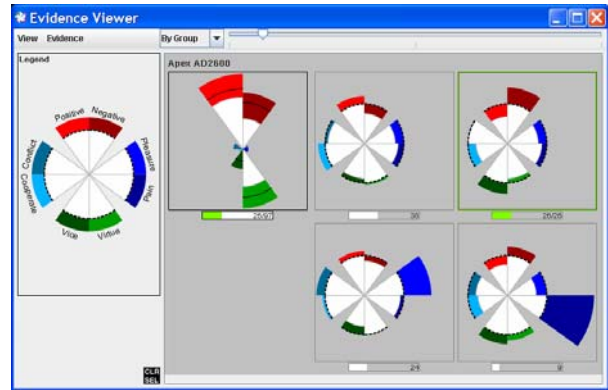


Figure 8. Summary of Apex products with sub-clusters showing deviations.



Figure 9. Thematic distribution of reviews for one product (Apex).

Looking at the thematic distribution among the Apex documents shows topics that dominate its reviews (Figure 9).

We can examine the affect across these various clusters. Figure 10 shows the comparison of the “service” cluster to the “dvd player picture” cluster. This graphic demonstrates that documents with “service” as a main theme tend to be much more negative, while documents with “picture” as a main theme are much more positive.

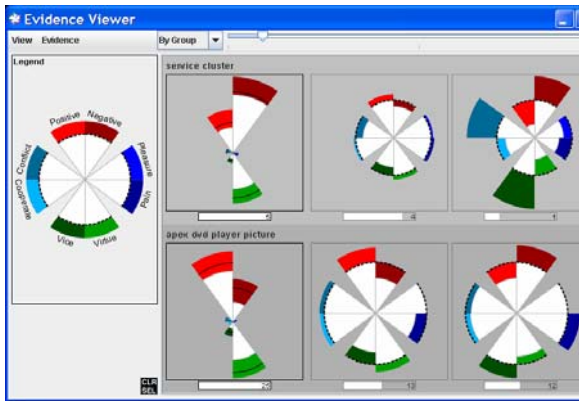


Figure 10. Affect summary and variation for “service” cluster and “picture” cluster.

The visualization tool includes a document viewer so that any selection of documents can be reviewed. For example, a user may be interested in why the “service” documents tend to be negative, in which case they can review the original reviews. The doc viewer, shown in Figure 11, can be used at any stage in the process with any number of documents selected. Individual documents can be viewed by clicking on a document title in the upper portion of the doc viewer.

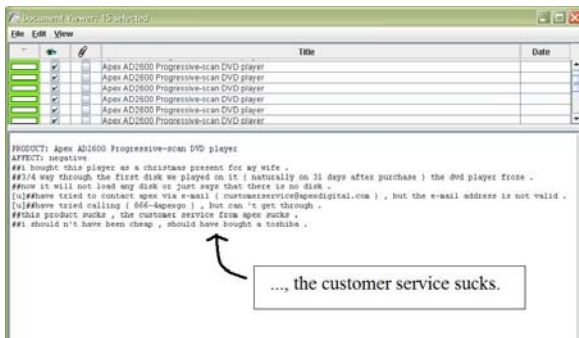


Figure 11: The Doc Viewer.

In this case study, we have illustrated the usefulness of visualizing the emotional content of a document collection. Using the tools presented here, we can summarize the dataset by saying that in general, the customer reviews are positive (Figure 5), but reviews for some products are more positive than others (Figures 6 and 7). In addition to the general content of the reviews, we can narrow our focus to the features contained in the reviews. We saw that while reviews for Apex are generally positive (Figure 8), reviews about Apex “service” tend to be much more negative than reviews about Apex “picture” (Figure 10).

## 5 Evaluation

IN-SPIRE is a document visualization tool that is designed to explore the thematic content of a

large collection of documents. In this paper, we have described the added functionality of exploring affect as one of the possible dimensions. As an exploratory system, it is difficult to define appropriate evaluation metric. Because the goal of our system is not to discretely bin the documents into affect categories, traditional metrics such as precision are not applicable. However, to get a sense of the coverage of our lexicon, we did compare our measurements to the hand annotations provided for the customer review dataset.

The dataset had hand scores (-3-3) for each feature contained in each review. We summed these scores to discretely bin them into positive (>0) or negative (<0). We did this both at the feature level and the review level (by looking at the cumulative score for all the features in the review). We compared these categorizations to the scores output by our measurement tool. If a document had a higher proportion of positive words than negative, we classified it as positive, and negative if it had a higher proportion of negative words. Using a chi-square, we found that the categorizations from our system were related with the hand annotations for both the whole reviews (chi-square=33.02, df=4, p<0.0001) and the individual features (chi-square=150.6, df=4, p<0.0001), with actual agreement around 71% for both datasets. While this number is not in itself impressive, recall that our lexicon was built independently of the data for which it was applied. We also expect some agreement to be lost by conflating all scores into discrete bins, we expect that if we compared the numeric values of the hand annotations and our scores, we would have stronger correlations.

These scores only provide an indication that the lexicon we used correlates with the hand annotations for the same data. As an exploratory system, however, a better evaluation metric would be a user study in which we get feedback on the usefulness of this capability in accomplishing a variety of analytical tasks. IN-SPIRE is currently deployed in a number of settings, both commercial and government. The added capabilities for interactively exploring affect have recently been deployed. We plan to conduct a variety of user evaluations *in-situ* that focus on its utility in a number of different tasks. Results of these studies will help steer the further development of this methodology.

## 6 Conclusion

We have developed a measurement and visualization approach to affect that we expect to be useful in the context of the IN-SPIRE text analysis toolkit. Our innovations include the flexibility of the lexicons used, the measurement options, the bootstrapping method and utility for lexicon development, and the visualization of affect using rose plots and interactive exploration in the context of an established text analysis toolkit. While the case study presented here was conducted in English, all tools described are language independent and we have begun exploring and creating lexicons of affect bearing words in multiple languages.

## References

- A. Aue. & M. Gamon. 2005. Customizing Sentiment Classifiers to New Domains: a Case Study. Submitted RANLP.
- S. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, and R.A. Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the Society for Information Science*, 41(6):391–407.
- J. M. DiMicco, V. Lakshmiopathy, A. T. Fiore. 2002. Conductive Chat: Instant Messaging With a Skin Conductivity Channel. In *Proceedings of Conference on Computer Supported Cooperative Work*.
- D. G. Feitelson. 2003. Comparing Partitions with Spie Charts. *Technical Report 2003-87*, School of Computer Science and Engineering, The Hebrew University of Jerusalem.
- E. Hetzler and A. Turner. 2004. Analysis Experiences Using Information Visualization. *IEEE Computer Graphics and Applications*, 24(5):22-26, 2004.
- M. Hu and B. Liu. 2004. Mining Opinion Features in Customer Reviews. In *Proceedings of Nineteenth National Conference on Artificial Intelligence (AAAI-2004)*.
- H. Lieberman, H. Liu, P. Singh and B. Barry. 2004. Beating Common Sense into Interactive Applications. *AI Magazine* 25(4): Winter 2004, 63-76.
- B. Liu, M. Hu and J. Cheng. 2005. Opinion Observer: Analyzing and Comparing Opinions on the Web. *Proceedings of the 14th international World Wide Web conference (WWW-2005)*, May 10-14, 2005: Chiba, Japan.
- H. Liu, T. Selker, H. Lieberman. 2003. Visualizing the Affective Structure of a Text Document. *Computer Human Interaction*, April 5-10, 2003: Fort Lauderdale.
- M. Mandic and A. Kerne. 2004. faMailiar—Intimacy-based Email Visualization. In *Proceedings of IEEE Information Visualization 2004*, Austin Texas, 31-32.
- B. Pang and L. Lee. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings of the 42nd ACL*, pp. 271-278, 2004.
- Y. Qu., J. Shanahan, and J. Weibe. 2004. Exploring Attitude and Affect in Text: Theories and Applications. Technical Report SS-04-07.
- E. Riloff and R. Jones. 1999. Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping. *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)* pp. 474-479.
- H. Scheffé. 1999. *The Analysis of Variance*, Wiley-Interscience.
- P. Stone. 1977. Thematic Text Analysis: New Agendas for Analyzing Text Content. In *Text Analysis for the Social Sciences*, ed. Carl Roberts, Lawrence Erlbaum Associates.
- A. Tat and S. Carpendale. 2002. Visualizing Human Dialog. In *Proceedings of IEEE Conference on Information Visualization, IV'02*, p.16-24, London, UK.
- P. Turney and M. Littman. 2003. Measuring Praise and Criticism: Inference of Semantic Orientation from Association. *ACM Transactions on Information Systems (TOIS)* 21:315-346.
- H. Wainer. 1997. *A Rose by Another Name.* Visual Revelations, Copernicus Books, New York.
- H. Wang, H. Prendinger, and T. Igarashi. 2004. Communicating Emotions in Online Chat Using Physiological Sensors and Animated Text." In *Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'04)*, Vienna, Austria, April 24-29.
- J. Wiebe and Ellen Riloff. 2005. Creating Subjective and Objective Sentence Classifiers from Unannotated Texts." In *Proceedings of Sixth International Conference on Intelligent Text Processing and Computational Linguistics*.
- T. Wilson, J. Wiebe, and P. Hoffmann. 2005. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis." In *Proceeding of HLT-EMNLP-2005*.



# DocuBurst: Visualizing Document Content using Language Structure

Christopher Collins<sup>1</sup>, Sheelagh Carpendale<sup>2</sup>, and Gerald Penn<sup>1</sup>

<sup>1</sup>University of Toronto, Toronto, Canada; <sup>2</sup>University of Calgary, Calgary, Canada

---

## Abstract

*Textual data is at the forefront of information management problems today. One response has been the development of visualizations of text data. These visualizations, commonly based on simple attributes such as relative word frequency, have become increasingly popular tools. We extend this direction, presenting the first visualization of document content which combines word frequency with the human-created structure in lexical databases to create a visualization that also reflects semantic content. DocuBurst is a radial, space-filling layout of hyponymy (the IS-A relation), overlaid with occurrence counts of words in a document of interest to provide visual summaries at varying levels of granularity. Interactive document analysis is supported with geometric and semantic zoom, selectable focus on individual words, and linked access to source text.*

Categories and Subject Descriptors (according to ACM CCS): Document And Text Processing [I.7.1]: Document and Text Editing—Document Management; Computer Graphics [I.3.6]: Methodology and Techniques—Interaction Techniques; Information Storage and Retrieval [H.3.7]: Digital Libraries—User Issues

---

## 1. Introduction

‘What is this document about?’ is a common question when navigating large document databases. In a physical library, visitors can browse shelves of books related to their interest, casually opening those with relevant titles, thumbing through tables of contents, glancing at some pages, and deciding whether this volume deserves further attention. In a digital library (or catalogue search of a traditional library) we gain the ability to coalesce documents which may be located in several areas of a physical library into a single listing of potentially interesting documents. However, the experience is generally quite sterile: people are presented with lists of titles, authors, and perhaps images of book covers. In feature-rich interfaces, page previews and tables of contents may be browsable. If the library contents are e-books, users may even open the entire text, but will have to page through the text slowly, as interfaces are often designed to present a page or two at a time (to dissuade copying). Our goal in this work is to bring some of the visceral exploratory experience to digital libraries, to provide interactive summaries of texts which are comparative at a glance, can serve as decision support when selecting texts of interest, and provide entry points to explore specific passages.

Prompted by the ever increasing volume and open access to digital text, developing overviews of document content has been an active research area in information visualization for many years. However, reported works do not make use of existing richly studied linguistic structures, relying instead on simple statistical properties of documents (*e.g.*, [AC07]), or analytic methods such as latent semantic analysis (*e.g.*, [DFJGR05]), which can produce unintuitive word associations. The resulting visualizations provide detail on content without a consistent view that can be compared across documents. In DocuBurst, we provide a complement to these works: a visualization of document content based on the human-annotated IS-A noun and verb hierarchies of WordNet [Fel98] which can provide both uniquely- and consistently-shaped glyph representations of documents, designed for cross-document comparison (see Figure 1).

## 2. Related Work

### 2.1. Document Content Visualization

Visualizations of document content take two common forms: synoptic visualizations for quick overviews and visualizations specialized for discovering patterns within and between documents. Specialization in the type of document



and based on statistical measures whose meaning may not be readily apparent to a reader. Note that all visualizations that provide overviews of entire text suffer from screen real estate issues with large texts.

## 2.2. Graph Drawing

Radial graph-drawing techniques have been previously reported and serve as the basis of this work. Of particular interest are the semi-circular radial space-filling (RSF) hierarchies of Information Slices [AH98] and the focus + context interaction techniques of the fully circular Starburst visualization [SZ00]. The InterRing [YWR02] visualization expands on the interaction techniques for RSF trees, supporting brushing and interactive radial distortion. TreeJuxtaposer [MGT\*03] illustrates methods for interacting with very large trees, where nodes may be assigned very few pixels. We adapt techniques such as tracing the path from a node of interest to the root and performing interactive accordion expansion from this work.

## 3. Background on WordNet

Despite the growing dependence on statistical methods, many Natural Language Processing (NLP) techniques still rely heavily on human-constructed lexical resources such as WordNet [Fel98]. WordNet is a lexical database composed of *words*, *collocations*, *synsets*, *glosses*, and *edges*. *Words* are literally words as in common usage. A *collocation* is a set of words such as “information visualization” which are frequently collocated and can be considered a unit with a particular definition. For the purposes of this paper, we will use *words* to refer to both *words* and *collocations* — they are treated equally in the visualization. Sets of synonymous *words* and *collocations* are called *synsets*. *Glosses* are short definitions that the words in a synset share, thus they are definitions of synsets. An edge in WordNet represents a connection between synsets.

*Synsets* are the most important data unit in WordNet. Throughout this paper, we will refer to *words* in single quotes (e.g. ‘thought’), and synsets using a bracketed set notation (e.g. {*thought*, *idea*}). A *word* may be a member of multiple *synsets*, one for each sense of that word. Word senses are ranked, either by order of familiarity (a subjective judgement by the lexicographer) or, in some cases, by using a synset-tagged reference corpus to provide numerical relative frequencies.

Synsets in WordNet are connected by many types of edges, depending on the part of speech (noun, verb, etc.). WordNet contains 28 different types of relations, but the most widely used part of WordNet is the hyponymy (IS-A) partial order. An example of hyponymy is {*lawyer*, *attorney*} IS-A {*professional*, *professional person*}. When traversing this graph, we remove any cycles (they are very rare) by taking a depth-first spanning tree at the user-selected root. In this work we focus on the noun hyponymy relationships

in English WordNet (v2.1), rooted under the synset {*entity*} having 73,736 nodes (synsets) and 75,110 edges, and a maximum depth of 14. Verb hyponymy is also supported — that hierarchy is smaller and takes a more shallow, bushier form. In addition, there is no single “root” verb. The visualizations produced can be generalized to any partial order of a lexicon.

## 3.1. WordNet Visualization

Many interfaces for WordNet exist, the most popular of which is the text-based WordNet Search which is part of the publicly available WordNet package. With the exception of the work of Kamps [KM02], the existing interfaces for WordNet either provide for drill-down textual or graphical interaction with the data starting at a single synset of interest or provide path-tracing between two synsets e.g., [Alc04, Thi05]. We do not know of any visualization of WordNet that uses the graph structure to enhance a visualization of other data such as document content.

## 4. DocuBurst Visualization

The combined structure of WordNet hyponymy and document lexical content is visualized using a radial space-filling tree layout implemented with *prefuse* [HCL05]. Traversing the tree from center to periphery follows a semantic path of increasing specificity using the IS-A relation. In WordNet, synset members are ordered according to their polysemy count, which WordNet researchers call *familiarity*. Since more familiar words come first, we chose the first word in a synset as the node label. Label fonts are maximized, rotated to fit within the node, and overlap is minimized.

### 4.1. Linguistic Processing and Scoring

In order to populate a hyponymy hierarchy with word counts, several pre-processing steps are necessary. Starting with raw text, we subdivide the text into *tiles* based on the pre-existing structure, such as section headings. If no structure is detectable, we break the text into roughly coherent topic segments using a segmenter [Cho00]. For each tile, we label parts of speech (NOUN, VERB, etc.) [Bri93]. Nouns and verbs are then extracted and stemmed (e.g., books → book, going → go) using a morphological processor [Did03]. Punctuation is omitted. If short word sequences, noted in WordNet, are found in the document, the words are combined into a collocation, and treated as a single word.

Next we look up in which WordNet synsets the (*word*, *part-of-speech*) pairs occur. Because pairs usually occur in multiple synsets, we do not perform word sense disambiguation. Instead, we divide the word count amongst the available synsets. If WordNet supplies relative sense frequency information for a word, we use this to distribute the count. Otherwise, we distribute the count weighted linearly by sense rank. This results in weighted occurrence counts that are not

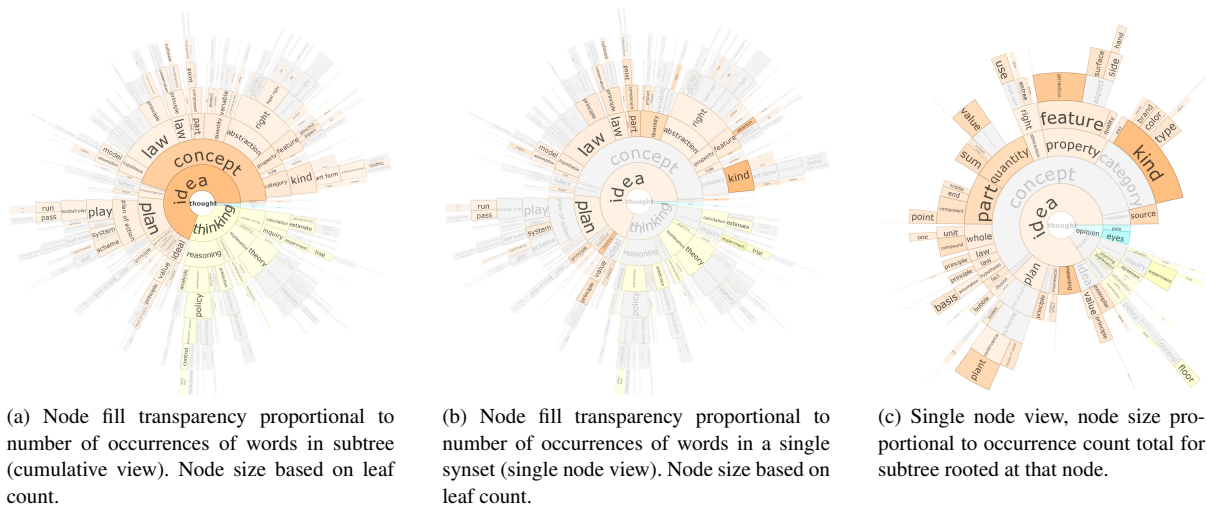


Figure 3: DocuBurst of a science textbook rooted at ‘thought’; node hue distinguishes the synsets containing ‘thought’.

integers, but the overall results more accurately reflect document content. By dividing the counts, we dilute the contribution of highly ambiguous terms. The full text of tiles and their associated (*word, part-of-speech, count*) triples are then read into the data structure of the visualization.

## 4.2. Visual Encoding

### Node Size

Within the radial tree, angular width can be proportional to the number of leaves in the subtree rooted at that node (*leaf count*) or proportional to the sum of word counts for synsets in the subtree rooted at that node (*occurrence count*). The leaf count view is dependent on WordNet and so is consistent across documents. The word count view maximizes screen space for synsets whose words actually occur in the document of interest, thus the shape, as well as node colouring, will differ across documents. Depth in the hyponymy tree determines on which concentric ring a node appears. The width of each annulus is maximized to allow for all visible graph elements to fit within the display space.

### Node Colour

It is possible to look at multiple senses of a word in one view. Views rooted at a single word contain a uniquely coloured subtree for each synset (sense) containing that word. In contrast, trees rooted at a single synset use a single hue. Since luminance variation in the green region of the spectrum is the most readily perceived, it is the first colour choice [Sto03, 30]. Gray is used for nodes with zero occurrence counts, since their presence provides a visual reminder of what words are not used.

Transparency is used to visualize relative word or synset

count. Similar to the concept of value, transparency provides a range of light to dark colour gradations, thus offering ordered [Ber83] and pre-attentive [War04] visuals. Highly opaque nodes have many occurrences; almost transparent nodes have few occurrences. Word senses that are more prominent in the document stand out against the more transparent context.

Two ways to visualize word occurrence are provided: single-node and cumulative. In the *single-node* visualization, only synset nodes whose word members occur in the document are coloured. In the *cumulative* view, counts are propagated up to the root of the tree. In both views, transparency is normalized so maximum counts achieve full opacity. When multiple documents are visualized, the cross-document maximum is used to set the scale. These modes support a gradual refinement of focus. The cumulative, or subtree, view uses the association of words into synsets and synsets into a hyponymy tree to aggregate counts for related concepts. Similar to the TreeJuxtaposer techniques for visualizing differences embedded deep in a large tree [MGT\*03], by highlighting the entire subtree containing the node, salient small nodes can be more easily located, even if hidden from view by a filter. The single-node view reveals precise concepts in the document and supports the selection of synsets whose word members appear in the document being analyzed. In addition, for a fully expanded graph, the single node view may highlight nodes that are otherwise too small to notice. The subtree and cumulative views are compared in Figure 3.

While transparency is an effective visual method for distinguishing large differences and trends, it is impossible to read exact data values using it. To facilitate the exact reading of synset occurrence counts for the selected text tiles, we provide a dynamic legend (see Figure 4).

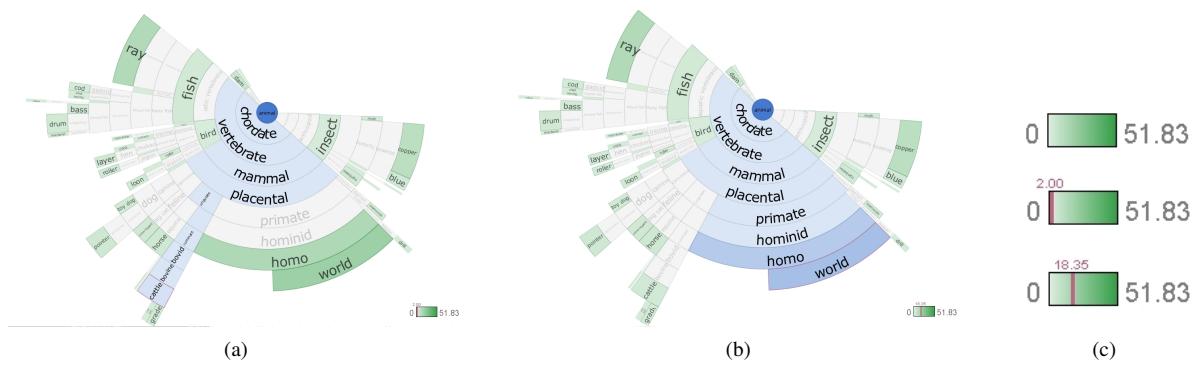


Figure 4: DocuBurst of a general science textbook rooted at  $\{animal\}$ . Single-node colouring and occurrence count sizing were used with zero-occurrence synsets hidden. Mouse hover point is revealed by blue trace-to-root colouring. (a) Synset  $\{cattle, cows, kine, oxen\}$  highlighted. (b) Synset  $\{world, human\ race, humanity, mankind, man, \dots\}$  highlighted. (c) Detail of the dynamic legend, showing, from top to bottom, no selection, selection from (a), and (b).

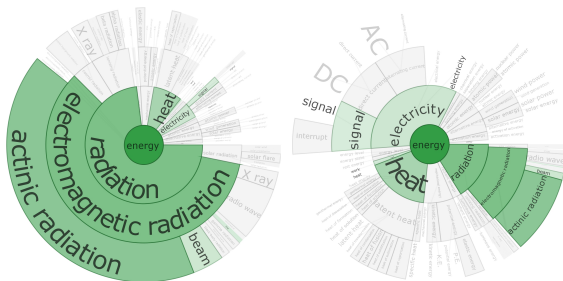


Figure 5: DocuBurst of a general science textbook rooted at  $\{energy\}$ . At right, mouse wheel interaction was used to shrink the angular width of the subtree rooted at  $\{radiation\}$  and expand the subtree under  $\{electricity\}$  exposing the previously illegible node  $\{signal\}$ .

## 5. Interaction

A root node can be a word, in which case its immediate children are the synsets containing that word. Alternatively the visualization can be rooted at a synset. Root nodes in the visualization are selectable by searching for either a word or synset of interest. Once a root is chosen, the visualization is populated with all its hyponyms.

As there are more than 70,000 English noun synsets in WordNet, techniques to abstract and filter the data are important. First, we provide a highlight search function which visually highlights nodes whose label matches any of the given search terms. *Highlight nodes* have a gold background and border, and a darker font colour, drawing attention to even the smallest of search results. The transparency of the highlight (gold) background is attenuated to the word occurrence counts so as to not disrupt this data-carrying value and to provide for stronger pop-out of search results with high occurrence counts.

Second, we implement a generalized fisheye view [Fur86] that collapses all subtrees which are more than a user-specified distance from the central root node. Changing this distance-based filter allows for a semantic zoom, creating visual summaries of varying specificity. The presence of non-zero word occurrence counts within collapsed subtrees is indicated by using the cumulative colouring, in which counts are propagated to the root. Optionally, all highlight nodes can be exempted from the distance filter (by increasing their *a priori* importance in the DOI function), effectively abstracting the graph to all synsets within a given distance from the root or highlight nodes (see Figure 1).

Double clicking on a node of interest restricts the visualization to the hyponyms of the node's synset; double right-clicking reverses this action by reloading the graph at the parent of the clicked node, thus providing bi-directional data navigation through the hyponymy relation. To create more space for the details of the children of a given synset, the angular width of a node and its subtree can be manually increased using the mouse wheel. This increase provides a radial detail-in-context view which causes the node's siblings to be correspondingly compressed. Changes to a node's angular width affect its children equally and its siblings in an inverse manner (see Figure 5).

The visualization can be based on selected subsections of the document. The initial view is based on all text tiles in the document, but a selection can limit the tiles from which counts are drawn. Unrestricted visual pan and geometric zoom of the display space are also supported, as well as a zoom-to-fit control to reset the pan and zoom to a best-fit for the currently visible tree. Rendering is dependent on the zoom factor: node borders are not rendered when the nodes are very small, and labels are not rendered when they would not be legible. All highlighting, navigation, and emphasis interactions are provided in real time.





details-on-demand provide a visual document summary, revealing what subset of language is covered by a document, and how those terms are distributed.

Initially motivated by the current lack of a digital equivalent of flipping through a book, this work leads well into an investigation of the DocuBurst technique to view the differences between two or more documents, which may be useful for plagiarism detection, document categorization, and authorship attribution. Existing digital library interfaces could be enhanced with arrays of DocuBurst icons, allowing comparison against one another or a baseline reference corpus to portray content in more pleasing and information-rich ways.

### Acknowledgements

Thanks to Ravin Balakrishnan for advice and guidance. Funding for this research was provided by NSERC, iCore, SMART Technologies, and NECTAR.

### References

- [AC07] ABBASI A., CHEN H.: Categorization and analysis of text in computer mediated communication archives using visualization. In *Proc. of the Joint Conf. on Digital Libraries* (2007), ACM, pp. 11–18.
- [AH98] ANDREWS K., HEIDEGGER H.: Information slices: Visualising and exploring large hierarchies using cascading, semi-circular discs. In *Proc. of IEEE Symp. on Information Visualization (InfoVis), Late Breaking Hot Topics* (1998), pp. 9–12.
- [Alc04] ALCOCK K.: WordNet relationship browser [online]. June 2004. Available from: <http://www.ultrasw.com/alcock/> [cited 20 February, 2006].
- [Bed00] BEDERSON B.: Fisheye menus. In *Proc. of the ACM Symposium on User Interface Software and Technology (UIST 2000)* (2000), ACM Press, pp. 217–226.
- [Ber83] BERTIN J.: *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press, 1983.
- [Bri93] BRILL E.: POS tagger. Software, 1993. Available from: [http://www.cs.jhu.edu/~brill/RBT1\\_14.tar.Z](http://www.cs.jhu.edu/~brill/RBT1_14.tar.Z).
- [Cho00] CHOI F. Y. Y.: Advances in domain independent linear text segmentation. In *Proc. of the 2000 Conference of the North American Chapter of the Association for Computational Linguistics* (2000), pp. 26–33.
- [DFJGR05] DECAMP P., FRID-JIMENEZ A., GUINNESS J., ROY D.: Gist icons: Seeing meaning in large bodies of literature. In *Proc. of IEEE Symp. on Information Visualization, Poster Session* (Oct. 2005).
- [Did03] DIDION J.: Java WordNet Library [online]. 2003. Available from: <http://jwordnet.sourceforge.net> [cited 28 August, 2005].
- [Dun93] DUNNING T.: Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics* 19, 1 (1993), 61–74.
- [DZG\*07] DON A., ZHELEVA E., GREGORY M., TARKAN S., AUVIL L., CLEMENT T., SHNEIDERMAN B., PLAISANT C.: Discovering interesting usage patterns in text collections: Integrating text mining with visualization. In *Proc. of the Conf. on Information and Knowledge Management* (2007).
- [FD00] FEKETE J.-D., DUFOURNAUD N.: Compus visualization and analysis of structured documents for understanding social life in the 16th century. In *Proc. of the Joint Conf. on Digital Libraries* (2000), ACM.
- [Fei08] FEINBERG J.: Wordle: Beautiful word clouds [online]. 2008. Available from: <http://www.wordle.net> [cited 2 December, 2008].
- [Fel98] FELLBAUM C. (Ed.): *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, USA, 1998.
- [Fur86] FURNAS G. W.: Generalized fisheye views. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems* (Apr. 1986), ACM Press, pp. 16–23.
- [HCL05] HEER J., CARD S. K., LANDAY J. A.: prefuse: a toolkit for interactive information visualization. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems* (Apr. 2005), ACM Press.
- [Hea95] HEARST M. A.: Tilebars: visualization of term distribution information in full text information access. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems* (1995), ACM Press, pp. 59–66.
- [HWMT98] HETZLER B., WHITNEY P., MARTUCCI L., THOMAS J.: Multi-faceted insight through interoperable visual information analysis paradigms. In *Proc. of the IEEE Symp. on Information Visualization* (Oct. 1998), pp. 137–144.
- [KM02] KAMPS J., MARX M.: Visualizing WordNet structure. In *Proc. of the 1st International Conference on Global WordNet* (2002), pp. 182–186.
- [MGT\*03] MUNZNER T., GUIMBRETIERE F., TASIRAN S., ZHANG L., ZHOU Y.: Treejuxtaposer: Scalable tree comparison using focus+context with guaranteed visibility. *ACM Transactions on Graphics* 22, 3 (2003), 453–462. SIGGRAPH 2003.
- [OBK\*08] OELKE D., BAK P., KEIM D. A., LAST M., DANON G.: Visual evaluation of text features for document summarization and analysis. In *Proc. of the IEEE Symp. on Visual Analytics Science and Technology (VAST)* (2008), pp. 75–82.
- [Pal02] PALEY W. B.: TextArc: Showing word frequency and distribution in text. In *Proc. of the IEEE Symp. on Information Visualization* (Oct. 2002), Poster, IEEE Computer Society.
- [Sto03] STONE M. C.: *A Field Guide to Digital Color*. AK Peters, Ltd., 2003.
- [SZ00] STASKO J., ZHANG E.: Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *Proc. of the IEEE Symp. on Information Visualization* (2000), pp. 57–65.
- [Thi05] THINKMAP: ThinkMap visual thesaurus, Apr. 2005. Available from: <http://www.visualthesaurus.com> [cited 10 April, 2005].
- [War04] WARE C.: *Information Visualization: Perception for Design*, 2nd ed. Morgan Kaufmann, 2004.
- [Wat02] WATTENBERG M.: Arc diagrams: Visualizing structure in strings. In *Proc. of the IEEE Symp. on Information Visualization* (2002).
- [WV08] WATTENBERG M., VIÉGAS F. B.: The word tree, and interactive visual concordance. *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization)* 14, 6 (Nov/Dec 2008), 1221–1229.
- [YWR02] YANG J., WARD M. O., RUNDENSTEINER E. A.: InterRing: An interactive tool for visually navigating and manipulating hierarchical structures. In *Proc. of the IEEE Symp. on Information Visualization* (2002), pp. 77–84.



# The Chinese Room: Visualization and Interaction to Understand and Correct Ambiguous Machine Translation

Joshua Albrecht<sup>1</sup>, Rebecca Hwa<sup>1</sup>, G. Elisabeta Marai<sup>1</sup>

<sup>1</sup>University of Pittsburgh, Department of Computer Science

---

## Abstract

We present The Chinese Room, a visualization interface that allows users to explore and interact with a multitude of linguistic resources in order to decode and correct poor machine translations. The target users of The Chinese Room are not bilingual and are not familiar with machine translation technologies. We investigate the ability of our system to assist such users in decoding and correcting faulty machine translations. We found that by collaborating with our application, end-users can overcome many difficult translation errors and disambiguate translated passages that were otherwise baffling. We also examine the utility of our system to machine translation researchers. Anecdotal evidence suggests that The Chinese Room can help such researchers develop better machine translation systems.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Information Visualization—Machine Translation

---

## 1. Introduction

The field of Machine Translation (MT) is concerned with developing methods for automating the task of translating between two natural human languages, such as Chinese and English. However, because this task is difficult even for skilled human translators, and requires a considerable amount of world knowledge that cannot be easily encoded in straightforward algorithms, the sentences produced by current MT systems are often difficult or impossible to understand. Consider the following example output:

```
"He utter eyes and not the  
slightest attention As leakage."
```

The resulting output is more accurately described as a jumble of words than an English sentence, even though the output was produced by one of the best MT systems freely available today [Goo08, NIS06]. The translation for the original sentence should have been:

```
"His eyes were wide apart; noth-  
ing in their field of vision es-  
caped."
```

Many researchers are working hard to improve MT directly by creating better algorithms and systems. We pursue an alternative solution – allow human users access to the significant amount of information available to an MT system, and let the user correct the MT output. This idea has been proposed as far back as 1980 by Martin Kay [Kay80], but that work and subsequent approaches have focused on improving the performance of professional translators. Our goal is to allow even users who are not bilingual to gain most of the information contained in the original source sentence.

In our approach, the human user relies on the machine to do the “symbol-pushing,” while the machine relies on the human’s world knowledge to guide the search for a correct translation. The assumption is that, although our intended users do not know the source language and may not be familiar with MT technologies, they have enough world knowledge and linguistic abilities in their native language to help them “decode” the disfluent MT output.

The process of language translation is complex and depends on different types of linguistic information; some of these information sources may contradict each other. Access to all this information could quickly overwhelm a potential

user, so it is essential that we design an effective system that presents the information visually, in a useful and understandable manner.

We present a prototype of a collaborative translation system, called *The Chinese Room*. The system visualizes ambiguous linguistic information about the unknown foreign language as it relates to the user's native language so that the user may gain an intuitive feel for what the source text might mean and thus overcome the mistakes made by the MT system.

Moreover, to better understand how users interact with the system and help MT researchers design stronger automated translation systems, we have designed an analysis module based on the timeline work of Plaisant et al. [PMR\*96]. A contribution of this work is that we demonstrate a novel domain application in which visualization helps machine translation research.

## 2. Background and Related Work

### 2.1. Background

In the process of translation, typical MT systems use a variety of linguistic resources — in terms of both data and tools. Each of these tools and data sources — briefly reviewed below — may introduce errors in the translation process.

In a first step, the source text is typically *segmented* into words; the task is not trivial for Chinese text, which contains no spaces naturally. As with all such automatic linguistic tools, errors are made during the segmentation, which can cause further errors later in the MT process.

Each sentence in the source text can also be *tagged* and *parsed*, in order to test conformability of the sentence to a logical grammar. The tagging step labels each word with a part-of-speech (POS) tag, such as noun, verb, or adjective. The POS-tagged sentence is then parsed; the parsing step produces a parse of the sentence — a tree structure showing the relationships among words within the sentence. Errors in the tagging and parsing steps can also propagate to the MT output.

Using the parsing information and *dictionaries*, the words are finally translated to the target language. Additional errors are possible at this stage. For example, the Chinese character pronounced *mei3* could mean either the noun "beautiful", or the noun "the United States." Choosing one definition over the other within a sentence can lead to very different machine translations: *He was responsive to beauty...*, as opposed to *He was sensitive to the United States...*

The output of the MT system is not only the translated sentence, but also a mapping between the words in each language. This mapping is called the *alignment*. In addition to the translation and alignment, some MT systems can also provide the *n-best* retranlations of a group of words. These retranlations are alternative translations generated by the

MT system, sorted in decreasing order of what the MT system considers to be their probability of being correct.

In addition to bilingual dictionaries, an MT system may consult resources such as *glosses*, *monolingual* or *bilingual corpora*. A gloss is a brief summary of a word's meaning, equivalent to the dictionary entry of that word, but only a word or two in length; it serves as a simple translation. A corpus (plural corpora) or text corpus is a large and structured set of texts typically used for statistical analysis. In more recent approaches, MT systems may also search the web for already-translated similar phrases.

### 2.2. Related Work

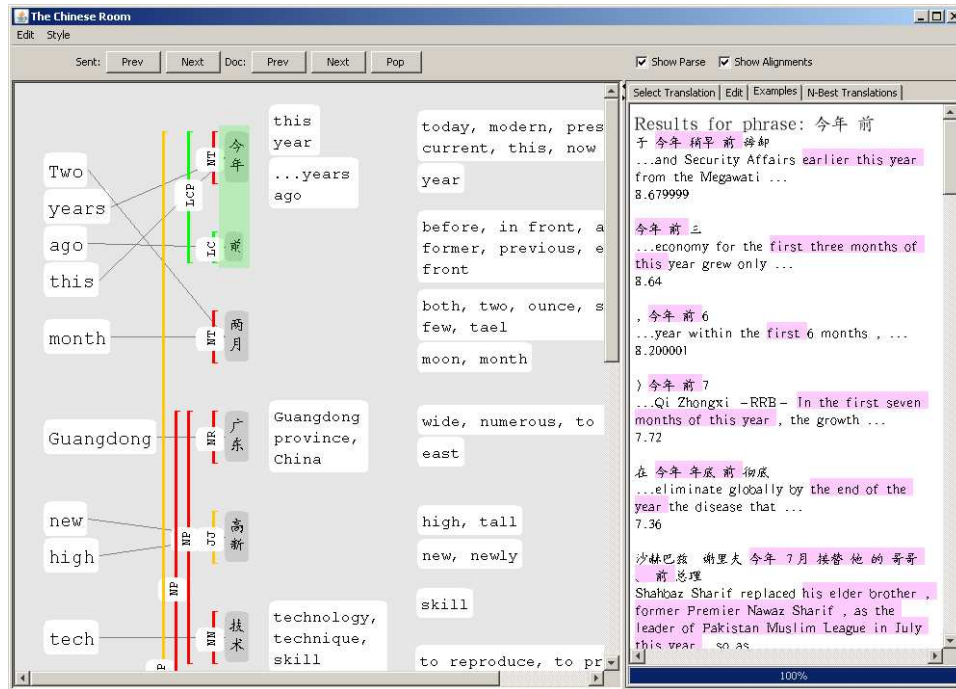
While there is a considerable body of knowledge on the design of MT systems, there has been surprisingly little work in the area of visualizing machine translation. Recently, a tool called DerivTool [DKC05] was created for the purposes of interacting with the core of an MT system. However, DerivTool's focus was on directly improving a specific MT system, and as such required in-depth knowledge of the particular MT system under analysis. Our application handles generic MT data and targets users who are not familiar with MT technologies.

The idea of leveraging human-computer collaborations to improve MT is not new; computer-aided translation, for instance, was proposed by Kay [Kay80]; research systems and commercial products have been successfully developed [Bow02,LFL00]. The focus of these efforts has been on improving the performance of professional translators. In contrast, *The Chinese Room* is targeted at users who cannot read the source text. Our objective is also related to that of cross-language information retrieval [ROL01] in that we want to help users to gain a deeper understanding of the information in the documents retrieved.

The name of our system was inspired by Searle's *Chinese Room* thought experiment [Sea80], although there are major differences between our system and Searle's description. Most notably, our users manipulate Chinese symbols by inserting their knowledge rather than purely operating based on instructions; nonetheless, the name was evocative in that our users require additional resources to process the input symbols.

Established methods exist for visualizing and interacting with the various components — from text to trees — that come into play in our machine translation application. Our overall design of *The Chinese Room* is based on the graphical design principles outlined by Tufte [Tuf90] and on the classic interaction principles of Card et al. [CMS99]. Our visualization and browsing of parse trees was further inspired by the work of Munzner et al. [MGT\*03].

Finally, the analysis module we designed in order to investigate how users collaborate with *The Chinese Room* builds on the timeline work of Plaisant et al. [PMR\*96]



**Figure 1:** The graphical environment consists of two main panes. The left pane is a workspace for exploring the sentence, while the right pane consists of multiple tabs that provide additional functionalities. The workspace displays the machine translation (leftmost column), source sentence, and alignments between them; the source sentence is annotated with its parse tree (colored brackets), its word and character glosses (two right columns). Currently displayed in the right pane is the example tab, showing the search results (highlighted in pink) for the selected Chinese phrase (highlighted in the left pane in green).

### 3. Methods

#### 3.1. Overview

In designing the prototype for *The Chinese Room*, we attempt to present the users with as many of the resources commonly used by MT systems as possible. Although many language-processing tools and multilingual resources are available as off-the-shelf packages, most are still imperfect. Finding the optimal way to integrate and display the possibly conflicting information from these resources is a challenging problem.

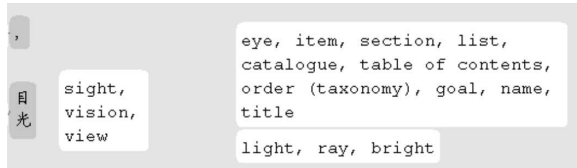
*The Chinese Room* consists of a visualization interface interconnected with five off-the-shelf text-processing modules: a machine translation and alignment module (the research version of Google's free Machine Translation service [Goo08]), a part-of-speech tagger (POS-tagger) and a parser module [KM03], a segmentation module (a by-product of Google's translation process), a custom glosses builder (we used the Chinese-English Translation Lexicon released by the Linguistics Data Consortium), and a custom information retrieval engine [Lem06], which allows the users to search large monolingual and bilingual corpora for approximate matches to difficult phrases. For corpora, we used the Federal Broadcast Information Service corpus and the Chi-

nese Gigaword corpus. We chose five widely used resources that were freely available and had a reasonably good performance. Additional less-common resources — such as translation phrase dictionaries, multiple dictionaries, or translation rules for phrases — could be added to the application; exploring the performance of alternative translation algorithms and additional resources in the context of our system goes beyond the scope of this paper. Each text-processing module provides linguistic information, while the visualization interface presents these linguistics resources to the users in an intuitive fashion, and also facilitates the user interaction.

An additional behind-the-scenes module (Section 3.4) allows the MT researchers to examine visually the MT corrections performed by users. The module is packaged with the application, but it is typically accessed only by the MT researchers.

#### 3.2. The Chinese Room Visual Display

Once the five text-processing modules process the source text, the resulting data must be displayed to the user in a way that is both simple enough to understand, and comprehensive enough for the task of understanding and correcting the

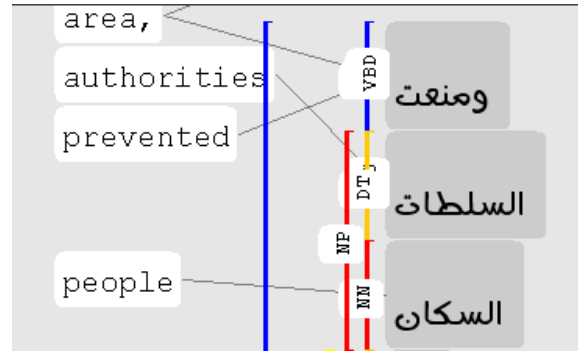


**Figure 2:** Source text segmentation: the ideograms for 'eye' and 'light' or 'ray' form the word 'sight'.

translation to be possible. The graphical display design was guided by iterative prototyping and feedback from both MT researchers and novice users. Early prototypes attempted to show the user simultaneous multiple views of the resources (for example, the document view, the sentence view, and the detail view — such as the alternative translations). However, feedback indicated that showing more than two views at a time was confusing the users. Later, more successful designs emphasized instead the translation task by showing the current translated sentence at all times in a left pane, while allowing the user to interactively view either details on demand or the document context in a second pane (Fig. 1). The left pane serves as a large workspace in which the user can interact with the translated text sentence by sentence; on the right pane are tabbed panels that give users access to information with additional context.

The left pane combines five sources of information – the most important and easier to use according to early feedback: the segmented source sentence, its translation, the alignment of the source and target sentence, the parse structure of the sentence, glosses for words and, in the Chinese case, glosses for characters. The text for both the initial machine translation and the dictionary definitions — indicated by user feedback as convenient resources — is displayed clearly in the white, rounded boxes. Text for the source sentence is shown in darker grey boxes, giving it less visual prominence because Chinese or Arabic characters are not directly useful to our users, who can't read them. The segmentation, which was useful to our users, is still readily apparent. For example, in Fig. 2 the users can see how the MT system built the word 'sight' from two ideograms, 'eye' and 'light'.

Alignments between the source words and the target words (Fig. 3) are shown in a dark grey color because they are often wrong or uninformative. The alignments allow the users to visually detect potential misalignments or poor word reordering. For instance, the automatic translation shown in Figure 1 begins: *Two years ago this month...* It is fluent but incorrect. The crossed alignments offer users a clue that "two" and "months" should not have been split up. English words in the machine translation are clustered together based on these alignments. The intuition is that alignments group words into likely phrasal units of translation, making it eas-



**Figure 3:** Source to target alignment showing the MT mappings between source words in Arabic and target words in English. POS-tags label the parse tree, although novice users tend to focus on the tree structure and at most the first letter of each label: NP (noun phrase), NN (single noun), VBD (verb past tense), DT (determiner) etc.

ier to see at a glance (a) how the sentence is structured, and (b) if any phrase looks out of place.

Glosses for words and characters are shown on the right side of the workspace pane. In the case of Chinese text, the placement of the word glosses presents a challenge because there are often alternative Chinese segmentations. We place glosses for multi-character words in the column closer to the source. When the user mouses over each definition, the corresponding characters are highlighted, helping the user to notice potential mis-segmentation in the Chinese. Dictionary definitions are organized based on the characters that they correspond to.

Finally, the source sentence is annotated with its parse structure. The design of the parse tree was challenging, since most users are not familiar with the concept of a parse tree. In such a tree, each node in the tree represents either a syntactic phrase (if it is an interior node), or a POS-tagged word (if it is a leaf node). We represent each syntactic node with a bracketed line that spans each of the child phrases and words. The brackets are color-coded into four major types (noun phrase, verb phrases, prepositional phrases, and other). Each node is also labeled with the name of the phrase so that the mapping between color and type does not need to be remembered by the users. Early feedback indicated the parse structure was useful in general when analyzing a phrase, while POS-labels were more useful to MT researchers than to novice users. Accordingly, in our pilot study the users were instructed to focus on the extent of brackets rather than their color-mapping and labels; future versions of the system will enable the users to turn off POS-labeling.

The right pane of the visual display can be used (1) as an overview tab that shows all sentences in the document; sen-

tences can be selected interactively to be worked on in the left pane; (2) as a notepad-like tab that allows direct editing of the machine translation; or (3) to display additional information (details-on-demand) such as alternative translations or similar phrases.

### 3.3. Interaction

When users encounter a problem that requires more information than is displayed by default, they have two options for exploration — searching for similar phrases, or requesting *n*-best retranslations for that phrase. Users can select a portion of the source string in order to search for similar phrases in a bilingual corpus; the search returns professional translations in similar contexts. Additionally, phrases from a large monolingual Chinese corpus are also returned together with their automatic translations. If the users wish to examine any of the translation pairs in detail, they can push it onto the sentence workspace. Finally, users can also request alternative translations from the MT system for selected sources phrases. The *N*-best alternatives are displayed.

In the left pane, users can zoom-in, collapse and expand the parse-tree brackets to keep the workspace uncluttered as they work through the source sentence. This action also indicates to the MT researcher which fragments held the user's focus. Highlighting either the original Chinese word or the definition in the left pane will show the matching definition or characters, respectively. Finally, all English elements can be edited and dragged around the screen, allowing the user to tangibly interact with the sentence, and consider alternative target-word orderings and thus alternative translation possibilities.

### 3.4. Visual Analysis

One of the issues of prime interest to the MT researchers was investigating how exactly people use *The Chinese Room*. What resources and strategies did they use? To answer such questions, and in collaboration with the MT researchers, we created a timeline visualization for each trial; a trial is defined as one document being worked on by a single user. The timeline diagrams provide a nice, simple description of the user behavior during the trial.

The analysis window (Fig. 4) is split horizontally into two main white panes. The MT researcher can load one trial — i.e., document/user pair — in each pane. The two panes allow for easier comparison of the trial pairs.

Each pane shows the user's actions for a particular document. Vertical lines of color represent the user actions on a specific sentence within a document; each horizontal line corresponds to a sentence within the document, and the *x*-axis is mapped to time. Each color represents a different action: editing, document context view, search results, alternate translation, or *other* — for example, examining an

example sentence in detail. Thus, colors represent which resources were used, and when.

The analysis window also contains two toggle buttons. The 'Type' button controls whether the plot boxes are showing cumulative action or the timeline of the actions. The 'Connected' button controls whether the time scale of the two boxes is scaled separately or is locked together. We note that the analysis module was designed with the goal of comparing the behavior of a few users at a time, which was a good fit with the exploratory nature of the present study. The module would likely have to be revised to fit potential larger-scale studies.

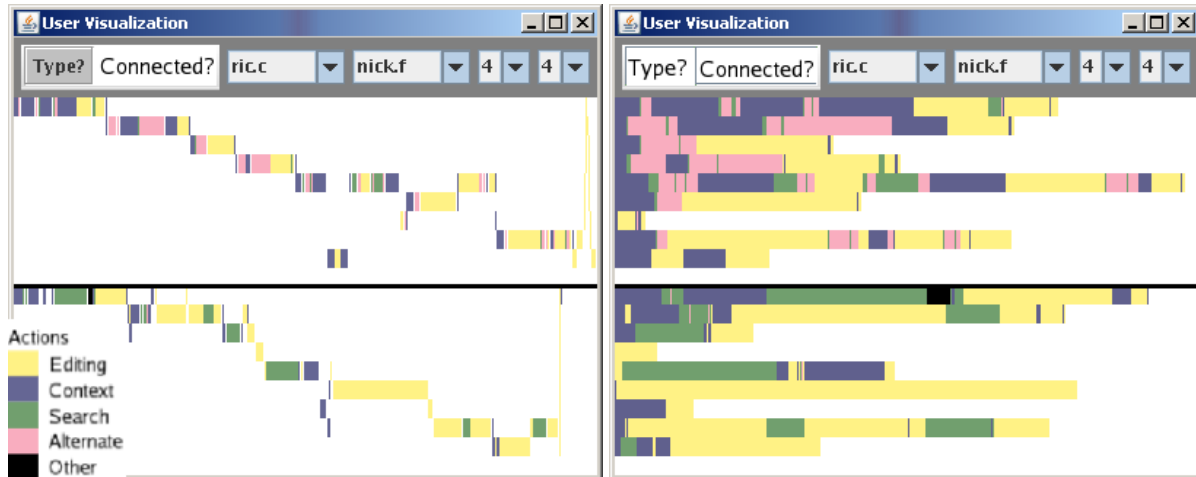
## 4. Evaluation and Results

To evaluate the MT impact of *The Chinese Room*, we have conducted a pilot experiment. We asked eight non-Chinese speakers to correct the machine translations of four short Chinese passages, approximately ten sentences long each. While both the participant pool and the dataset (184 participant-corrected sentences) are relatively small, they allowed the MT researchers to perform quantitative and qualitative assessments while controlling for user backgrounds and experiences (see [AHM09] for a detailed analysis).

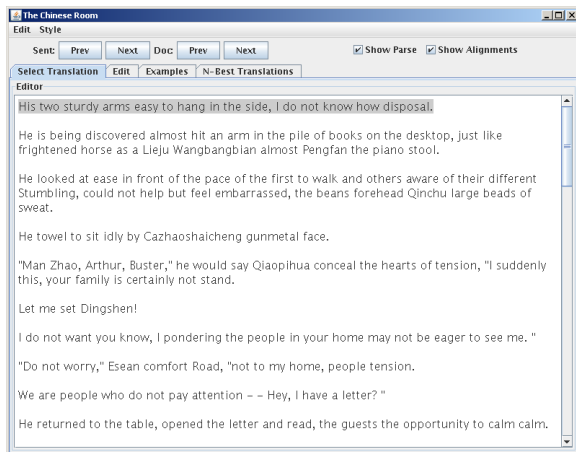
Each participant was instructed to (a) correct the translations for one news article and one fiction passage using all the resources made available by *The Chinese Room* and (b) correct the other two passages without *The Chinese Room*. To keep the experimental conditions as similar as possible, for task (b) we provided the users with a restricted version of the graphical interface (Fig. 5) in which all additional functionalities except for the Document View Tab were disabled. Half of the users began with task (b), and the other half with task (a). Thus, every passage received four sets of corrections made collaboratively with the system, and four sets of corrections made based solely on the participants' internal language models.

The participants were asked to complete each passage within one session, without further time constraints. Within a passage, the users could work on the sentences in any arbitrary order. They could also elect to "pass" any part of a sentence if they found it too difficult to correct. Timing statistics were automatically collected. We conducted a short exit interview with each participant at the end of the session.

The corrected translations were evaluated by two bilingual speakers ("judges"). The judges were presented with the original source text as well as the parallel English text for reference. Each judge was then shown a set of candidate translations: the original MT output, an alternative translation by a bilingual speaker, and corrected translations by the participants, in a randomized order. Since the human corrected translations are likely to be fluent, we have instructed the judges to concentrate more on the adequacy of the meaning conveyed. They were asked to rate each sentence on an



**Figure 4:** Two views of the visual analysis window, showing the actions of two users (*ric* and *nick*) as they worked on the same document 4, both using the *Chinese Room*. Left: action timeline view: right: cumulative action view. The MT researcher quickly saw that the first user (*ric*) used a wide variety of resources, while the second user relied almost exclusively on searching for similar examples. Both users took a similar approach of working their way through each example sentence one by one without skipping around much, and relying instead on local context.



**Figure 5:** The interface for users who are correcting translations without the *Chinese Room*; they have access to the document view, but they do not have access to any of the other resources.

absolute scale of 1-10, where 9-10 means “The meaning of the Chinese sentence is fully conveyed in the translation”, and 1-2 means “The translation makes no sense at all.” To reduce the biases in the rating scales of different judges, we normalized the judges’ scores, following standard practices in MT evaluation [BFF\*03]; the normalization resulted in scoring in the [0,1] range.

Using *The Chinese Room*, the experiment participants

were able to improve on average the MT quality from 0.35 to 0.53, closing the gap between the MT and bilingual translations by 36.9% without knowing the source language (the average score of bilingual translations was 0.83). Table 1 shows example outputs from the participants.

Overall, the MT outputs contained enough errors that the participants were able to improve, to a small degree, the MT quality even without *The Chinese Room*, from 0.35 to 0.42. These differences are all statistically significant (using a paired t-test with >98% confidence). In general, participants who used *The Chinese Room* had more instances of large improvements than participants who made corrections without help.

In many cases, multiple users were able to identify a translation error without being able to fix it, but one or two users managed to understand the intended meaning by working with our system. As an upper-bound for the effectiveness of the system, we construct a combined “oracle” user out of all 4 users that used the interface for each sentence. The oracle user’s average score is 0.70; in contrast, an oracle of users who did not use the system is 0.54 (relative to the MT’s overall of 0.353 and the bilingual translator’s overall of 0.833). This suggests *The Chinese Room* affords a potential for human-human collaboration as well.

The higher quality of corrections did require the participants to put in more time. Overall, the participants took 2.5 times as long when they had access to *The Chinese Room* than when they did not. This may be partly because the participants have more sources of information to explore, and partly because the participants tended to “pass” on fewer

	Score	Translation
MT	0.336	He is being discovered almost hit an arm in the pile of books on the desktop, just like frightened horse as a Lieju Wangbangbian almost Pengfan the piano stool.
Without The Chinese Room	0.263	Startled, he almost knocked over a pile of book on his desk, just like a frightened horse as a Lieju Wangbangbian almost Pengfan the piano stool.
With The Chinese Room	0.778	He was nervous, and when one of his arms nearly hit a stack of books on the desktop, he startled like a horse, falling back and almost knocking over the piano stool.
Bilingual Translator	0.934	Feeling nervous, he discovered that one of his arms almost hit the pile of books on the table. Like a frightened horse, he stumbled aside, almost turning over a piano stool.

**Table 1:** Example translation corrected by the participants and their scores. In this example, the initial MT was badly jumbled, but the informed user was able to recover most of the meaning.

sentences. We note that the goal of this project is to improve the quality of machine translations, and not to minimize the task completion time.

During the exit interviews, the participants were asked: (a) to give an overall summary of each translated document; (b) about their overall satisfaction with the tool, including suggestions for improvement; (c) about the specific strategies they used to collaborate with the system. In general, summaries for news articles were accurate, while, unsurprisingly, summaries for story excerpts were less so (for example, one participant mistook a fragment from *Martin Eden* for a spy story). Participant experiences, as revealed through the exit interviews, were generally positive. Because the users felt like they understood the translations better, they did not mind putting in the time to collaborate with the system. Specific comments included: “happy to have it [the tool]”, “at least it gives one something to do when one is stuck”, “it was fun”, while the suggestions for improvement caught several minor bugs in the user interface. Novice users did not find the POS-tags particularly useful, although having access to the parse tree structure was considered helpful.

During the exit interviews, the participants were also asked to describe strategies that they developed for collaborating with the system. Their responses fall into three main categories:

- **Divide and Conquer:** Some users found the parse trees helpful in identifying phrasal units, for which they subsequently required  $N$ -best retranslations or example searches. For longer sentences, they used the constituent collapse feature to help them reduce clutter and focus on a portion of the sentence.
- **Example Retrieval:** Using the search interface, users examined the highlighted query terms to determine whether the MT system made any segmentation errors. Sometimes, they used the examples to arbitrate whether they should trust any of the dictionary glosses or the MT’s lexical choices. Typically, though, they did not attempt to inspect the example translations in detail.
- **Document Coherence and Word Glosses:** Users often referred to the document view to determine the context for the sentence they are editing. Together with the word

glosses and other resources, the context clues helped the users make better lexical choices than when they made corrections without the full system and relied on document coherence alone.

In general, users often accessed the document context view and requested alternative translations. About half as often they searched for similar examples and expanded or collapsed the right pane tree. The option of inspecting retrieved examples in detail (i.e., bring them up on the sentence workspace) was rarely used, perhaps because novice users experienced a greater degree of uncertainty than professional translators.

Figure 4 shows anecdotal evidence of how the visual analysis interface was useful to the MT researchers: two users are shown correcting the same document, both using *The Chinese Room*. In this particular example, the MT researcher can quickly see that the first user (ric) used a wide variety of resources, while the second user (nick) relied almost exclusively on searching for similar examples. Both users took a similar approach of working their way through each example sentence one by one, in a sequence, and relying primarily on the local context. The MT researcher also noticed that both users cared enough to put some effort into editing the final pass, even though this was the last document and they had been using the *Chinese Room* for several hours; in Fig 4, see the little “editing pass” at the end of the trial, where the users ensured that each sentence was what they wanted. The researcher noted: “If we were to look back at previous documents done by these users, I could see how their approach changed over time.”

In general, the most important thing the MT researchers learned from the analysis visualizations was a sense for how different each person’s approach to the problem was. As confirmed by the exit interviews, some people skipped all over, some relied on certain resources much more than others, while some would spend half the time on the first sentence, then do the other sentences very quickly. The researchers had hoped to be able to say more about the usage of various linguistic resources, but since the pilot users used the interface in very different ways, it was difficult to make generalizations. Perhaps with more detailed logging, or a larger

user pool, more commonalities could have been found. The researchers wished that we had recorded slightly more information, such as mouse clicks and movements, but on the other hand, they also felt they already had a lot of information about the user behavior.

## 5. Discussion and Conclusion

In this paper, we have shown that concepts from visualization and human computer interaction can have significant positive impacts on research in machine translation.

From our experiments, *The Chinese Room* seems promising as an end-application. Although the participants did not find all the translation errors using our current system, the corrections they made led to a significant improvement over the original machine translation. Moreover, our timeline analysis module helped researchers to understand what resources humans need to correct erroneous machine translations and how they interact with these resources to achieve this task. In the future, we plan to incorporate additional resources specific to a particular type of MT systems to help researchers further improve their systems. The timeline analysis may also lead to alternative interface designs. For example, although feedback from earlier prototypes indicates that simultaneous multiple views are confusing to the users, the timeline analysis shows that participants frequently switch between different views. Ultimately, the *Chinese Room* serves as a useful diagnostic tool to help designers of MT systems to verify ideas about what types of resources and data are useful for automatic translation.

One of the most interesting challenges in this work was finding ways to incorporate into the visual display the uncertainty inherent to the various machine translation sub-processes. In visualization, uncertainty is often ignored or treated as a binary quality. In knowledge discovery applications it is, however, important to show to the user not only that the value of a particular option is uncertain, but also the other options available. In some instances, by showing alternative options, our visualization enabled the users to unravel and repair sequences of three or four consecutive mistakes.

In conclusion, we have designed, implemented and tested *The Chinese Room*, a visualization interface that makes available a variety of linguistic resources to users in an intuitive display and facilitates their collaborations with these resources. By combining evidence from complementary information sources, users can infer alternative hypotheses based on their world knowledge and improve the overall translation. Experimental evidence suggests that the collaborative effort between human users and our system results in much improved translations than either the original MT or uninformed human edits. Moreover, in several instances, the quality of the corrected translation approached that of bilingual speakers. User inputs may be gathered for error analysis and system training for future MT development. Finally, this work demonstrates a novel domain application in


which visualization and interaction help machine translation research.

**Acknowledgments** This work has been supported by NSF IIS-0710695 and IIS-0745914, and by a University of Pittsburgh startup grant.

## References

- [AHM09] ALBRECHT J., HWA R., MARAI G. E.: Correcting automatic translations through collaborations between mt and monolingual target-language users. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (in press)* (2009). 5
- [BFF\*03] BLATZ J., FITZGERALD E., FOSTER G., GANDRABUR S., GOUTTE C., KULESZA A., SANCHIS A., UEFFING N.: *Confidence estimation for machine translation*. Tech. Rep. Natural Language Engineering Workshop Final Report, Johns Hopkins University, 2003. 6
- [Bow02] BOWKER L.: *Computer-Aided Translation Technology*. University of Ottawa Press, Ottawa, Canada, 2002. 2
- [CMS99] CARD S., MACKINLAY J., SHNEIDERMAN B.: *Readings in information visualization: Using vision to think*. Morgan Kaufman, San Francisco, 1999. 2
- [DKC05] DENEEFE S., KNIGHT K., CHAN H. H.: Interactively exploring a machine translation model. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions* (Ann Arbor, Michigan, June 2005), pp. 97–100. 2
- [Goo08] GOOGLE: Google machine translation service. <http://code.google.com/apis/translate/research>. 1, 3
- [Kay80] KAY M.: *The proper place of men and machines in language translation*. Tech. Rep. CSL-80-11, Xerox, 1980. Later reprinted in *Machine Translation*, vol. 12 no.(1-2), 1997. 1, 2
- [KM03] KLEIN D., MANNING C. D.: Fast exact inference with a factored model for natural language parsing. *Advances in Neural Information Processing Systems 15* (2003). 3
- [Lem06] LEMUR: Lemur toolkit for language modeling and information retrieval, 2006. The Lemur Project is a collaborative project between CMU and UMASS. 3
- [LFL00] LANGLAIS P., FOSTER G., LAPALME G.: Transtype: a computer-aided translation typing system. In *Workshop on Embedded Machine Translation Systems* (May 2000), pp. 46–51. 2
- [MGT\*03] MUNZNER T., GUIMBRETIERE F., TASIRAN S., ZHANG L., ZHOU Y.: Treejuxtaposer: scalable tree comparison using focus+context with guaranteed visibility. *ACM Trans. Graph.* 22, 3 (2003), 453–462. 2
- [NIS06] NIST: 2006 machine translation evaluation official results. <http://www.itl.nist.gov/iad/mig/tests/mt/>. 1
- [PMR\*96] PLAISANT C., MILASH B., ROSE A., WIDOFF S., SHNEIDERMAN B.: Lifelines: Visualizing personal histories. In *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems* (1996), ACM Press, pp. 221–227. 2
- [ROL01] RESNIK P. S., OARD D. W., LEVOW G.-A.: Improved cross-language retrieval using backoff translation. In *Human Language Technology Conference (HLT-2001)* (2001). 2
- [Sea80] SEARLE J.: Minds, brains, and programs. In *Behavioral and Brain Sciences* (1980), vol. 3:417, pp. 34–57. 2
- [Tuf90] TUFTE E. R.: *Envisioning Information*. Graphics Press, Cheshire, Connecticut, USA, 1990. 2





Interactive Visualization for  
Computational Linguistics

ESSLI 2009

## Instructors

2

- Sheelagh Carpendale (sheelagh@ucalgary.ca)  
Associate Professor, University of Calgary
- Gerald Penn (gpenn@cs.utoronto.ca)  
Associate Professor, University of Toronto
- TA: Christopher Collins (ccollins@cs.utoronto.ca)  
PhD Candidate, University of Toronto

## Tutorial Objectives

*An understanding of the importance  
and applicability of information  
visualization techniques to  
computational linguistics research;*

*Knowledge of the basic principles of information visualization theory;*

*The ability to identify appropriate visualization software and techniques that are available for immediate use and for prototyping;*

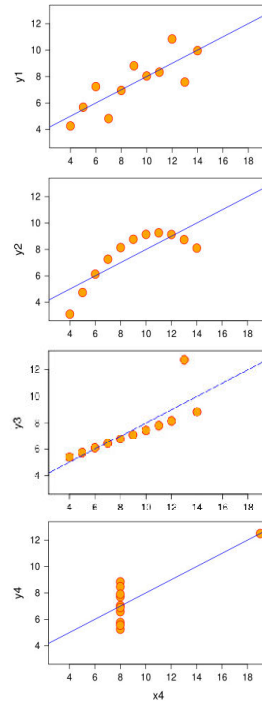
*A working knowledge of  
research to date in the area of  
linguistic visualization.*

*The Power of Visualization*

# Anscombe's Quartet

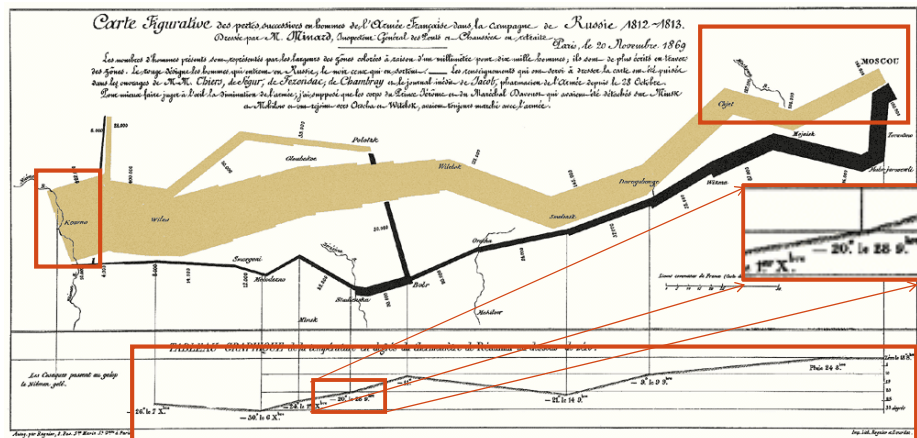
9

I		II		III		IV	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89



# Example: Movements of the French Army

10



temperature, date    number    location, direction

Minard, 1861; Tufte, 2001

# London Cholera Deaths

11

50 0 50 100 150 200  
Yards  
X Pump • Deaths from cholera



John Snow (1855) On the Mode of Communication of Cholera.

What is *information  
visualization*?

12

*Observations Jupiter*  
1610

2. J. Jovis	○ * *
30. Jovis	* * ○ *
2. Jovis	○ * * *
3. Jovis	○ * * *
3. Jovis	* ○ *
7. Jovis	* ○ * *
6. Jovis	* * ○ *
8. Jovis	* * * ○
10. Jovis	* * * ○ *
11.	* * ○ *
12. H. Jovis	* ○ *
17. Jovis	* * ○ *
14. Jovis	* * * ○ *

Galileo's notebook on Jupiter

Alan Turing's sketch of analysis process for a 25-letter Enigma Cipher Text.  
Courtesy of King's College, Cambridge and the UK National Cataloguing Unit for the Archives of Contemporary Scientists

# A Sentence

“Mattie was here last evening, and we sat on the front door stone, and talked about life and love, and whispered our childish fancies about such blissful things -- the evening was gone so soon, and I walked home with Mattie beneath the silent moon, and wished for you, and Heaven.”

Emily Dickinson's Letters to Susan Gilbert

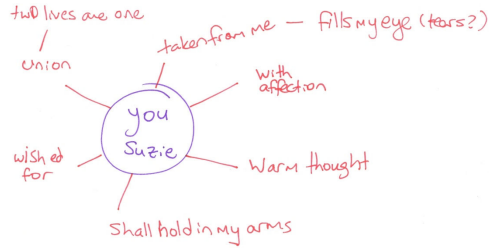
# A Single Letter

15

Early June, 1852

They are cleaning house today, Susie, and I've made a flying retreat to my own little chamber, where with affection, and you, I will spend this my precious hour, most precious of all the hours which dot my flying days, and the one so dear, that for it I barter everything, and as soon as it is gone, I am sighing for it again.

I cannot believe, dear Susie, that I have stayed without you almost a whole year long; sometimes the time seems short, and the thought of you as warm as if you had gone but yesterday, and again if years and years and trod their silent pathway, the time would seem less long. And now how soon shall I have you, shall hold you in my arms; you will forgive the tears, Susie, they are so glad to come that it is not in my heart to reprove them and send them home. I don't know why it is -- but there's something in your name, now you are taken from me, which fills my heart so full, and my eye, too. It is not that the mention grieves me, no, Susie, but I think of each "sunnyside" where we have sat together, and lest there be no more, I guess is what makes the tears come. Mattie was here last evening, and we sat on the front door stone, and talked about life and love, and whispered our childish fancies about such blissful things -- the evening was gone so soon, and I walked home with Mattie beneath the silent moon, and wished for you, and Heaven. You did not come, Darling, but a bit of Heaven did, or so it seemed to us, as we walked side by side and wondered if that great blessedness which may be our's sometime, is granted now, to some. Those unions, my dear Susie, by which two lives are one, this sweet and strange adoption wherein we can but look, and are not yet admitted, how it can fill the heart, and make it gang wildly beating, how it will take us one day, and make us all it's own, and we shall not run away from it, but lie still and be happy!



*[The text in this block is extremely small and illegible, appearing to be a list of references or a detailed index.]*

16



my dear 'Oliver,' how chipper  
 any of us have seen  
 seen you?" and "I hope  
 absent, and Emily devises punishment: "

You deserve, let me see;

Austin, I am keen, but  
 something of a fox, but  
 several weeks later beginning, "Do

Nothing -- I was aware that

was at an end -- ... So  
 sustained by that dear sister  
 to be sisters, when indeed

is no "silence" there -- so  
 here, Susie -- on purpose for

far before it gets to

... ..

you must be since any of  
 ?" and "I hope you have  
 you have been made happy." In  
 you deserve, let me see; you

you deserve hot irons, and Chinese

you are a good deal keener  
 you are more of a hound  
 you want to hear from me, Austin

you had been in correspondence

you will not suspect me of  
 you will never again be lonely  
 you were alone!  
 you do not hear the wind  
 you differ from bonnie "Alice." I  
 you because this "sweet silver moon"

you -- and then you never told

17 Several Dickinson Letters

Nora Visualization: emily-fulltext.nora

File Data Views Analysis

FEATURE	RATIO	ID	hot_prob	title	take
her	2.23	195	1.97	Least Bee that / Brew - / A Honey's Wo...	
my	1.98	196	2.28	I must wait / a few Days	
you	1.98	197	-1.2	Has All - a / codicil?	
susan	1.98	198	11.99	To take away our / Sue	
me	1.98	199	1.13	The Leaves like / Women interchange	
last	1.82	200	4.33	I send My / Own, two answers -	
sister	1.82	201	0	Success is counted sweetest	
take	1.82	202	9.16	Dear Sue - / It is / sweet you / are better.	
woman	1.64	203	4.43	Dear Sue - / I'm thinking / on that other...	
sue	1.64	204	0	Perception of an / object costs	
though	1.64	205	8.39	My Sue - / Loo and / Fanny will come /	
have	1.42	206	-0.23	A fresh / Morning	
god	1.42	207	3.21	The Bumble of a Bee - / A Witchcraft, yi...	
ill	1.42	208	11.36	Dear Sue - / With the / Exception of /	
heart	1.42	209	-1.9	The things of / which we want / the proof	
she	1.42	210	1.73	Mama and / Sister might / like a flower	
fit	1.42	211	-2.37	Now I lay / thee down to / Sleep	
believe	1.42	212	-1.27	Best Witchcraft / is Geometry / To a m...	
gone	1.42	213	8.95	Will my great / Sister accept	
only	1.42	214	0.46	"Egypt - thou / knew'st" -	
at	1.13	215	0.17	Please Excuse / Santa Claus	
face	1.13	216	0	For largest Woman's / Heart I knew -	
remem...	1.13	217	1.88	Thank Sue, but / not tonight.	
own	1.13	218	0	Susan - I dreamed / of you	
eden	1.13	219	1	Lest any doubt / that we are glad	
back	1.13	220	3.62	"For Brutus, / as you know"	
doubt	1.13	221	0.06	Shall I cannot be / altered.	

To take away our  
 Sue leaves but a  
 lower world, her  
 firmamental quality  
 our more familiar  
 stay.

It is not Nature -  
 dear, but those  
 that stand for  
 Nature.

The Bird would be  
 a soundless thing  
 without Expositor.  
 Come Home and  
 see you Weather.  
 The Hills are full  
 of Shawls, and I  
 am going every

User Rating  
 Not Hot 0 0 0 0 0 Hot Unrated

Predicted Rating  
 Not Hot Hot

18 Plaisant et al., NORAVIS, JCDL 2006

# Externalization

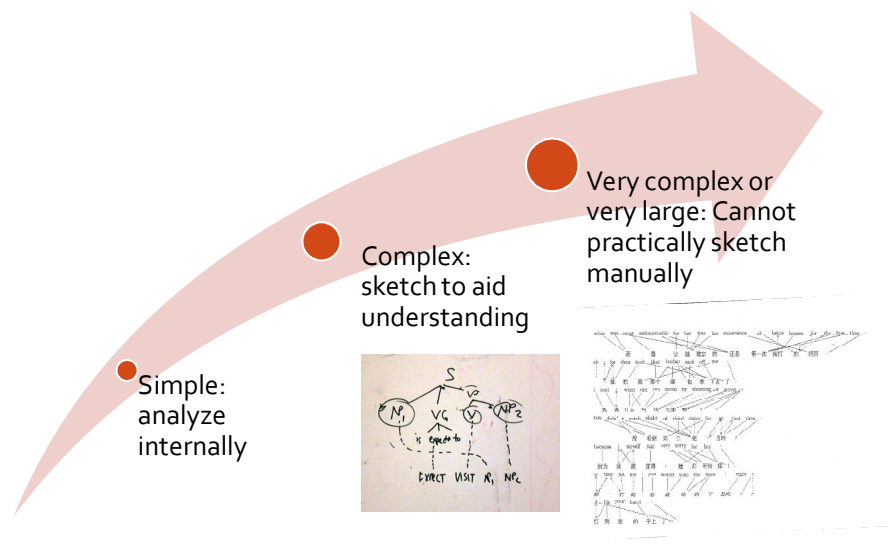
19

- External cognition is the interaction between internal and external representations when performing cognitive tasks.
- Computational offloading is the extent to which external representations can reduce the amount of cognitive effort to solve a problem.

Yvonne Rogers, New Theoretical Approaches for Human-Computer Interaction, 2004.

# Complexity Brings Externalization

20



## Augmented Cognition

21

“The power of the unaided mind is highly overrated. Without external aids, memory thought, and reasoning are all constrained. But human intelligence is highly flexible and adaptive, superb at inventing procedures and objects that overcome its own limits. The real powers come from devising external aids that enhance cognitive abilities.”

Norman, 1993, p. 43

## Information Visualization

22

“Information visualization is the use of computer-supported interactive visual representations of abstract data to amplify cognition.”

... a form of external cognitive aid.

Card, 1999

# Functions of Visualization

23

- Recording information
  - ▣ Tables, blueprints, satellite images
- Processing information
  - ▣ needs feedback and interaction
- Presenting information
  - ▣ share, collaborate, revise
  - ▣ for oneself, for one's peers and to teach
- Seeing the unseen

Carpendale

24

## Information Visualization Basics

# Creating a Visualization

25

1. Understand a **system of related information** and **tasks**.
2. Create a **mapping** from the **data** (digital representation) to a **visual representation**.
3. **Present** this visual representation on the computer screen.
4. Provide **methods of interacting** with this **visual representation** that can include methods for varying the **presentation** and methods for varying the **representation**.
5. **Verify** the usefulness of the **representation**, the way it is **presented** and/or and its **interaction** methods.

Carpendale

## *Visual Variables*

26

# References

27

- Slides in this section are based on:

*Semiology of Graphics*, Jacques Bertin, 1983  
translation of *Semiologie Graphique*, 1967

- With some extensions

# Representation

- A representation is
  - a formal system or mapping by which the information can be specified (D. Marr)
  - a sign system in that it stands for something other than its self.
- for example: the number thirty-four
  - decimal: 34,
  - binary: 100010,
  - roman: XXXIV
- different representations reveal different aspects of the information
  - decimal: counting & information about powers of 10,
  - binary: counting & information about powers of 2,
  - roman: counting & adding and subtracting
- presentation
  - how the representation is placed or organized on the screen

34, 34, 34

# Representations

- Solving a problem simply means representing it so as to make the solution transparent ... (*Simon, 1981*)
- Good representations
  - ▣ allow people to *find* relevant information
    - information may be present but hard to find
  
  - ▣ allow people to *compute* desired conclusions
    - computations may be difficult or “for free” depending on representations

# Creating Visual Representations

30

- To communicate with words we first need to know phonemes, the letters and how they combine to create words
  - ▣ note that phonemes and letters are “meaningless” in themselves
  
- are there corresponding visual units?
  - ▣ there is still considerable debate on this subject
  
- in the meantime, we will look at the practical approach of Jacques Bertin on how we can create visual representations that can be understood.

# Disclaimer

31

- **Bertin considers:**
  - printable, on white paper,
  - visible at a glance
  - reading distance of book or atlas
  - normal and constant lighting
  - readily available graphic means
- Good as a guideline, we need to consider context and medium of visualization use.

# Where do we start?

32

- With marks!
  - for us, pixels
- *Visual Variables*: how can we vary marks?
  - by where we place them
  - by how we place them
  - by their visual characteristics



# The Plane

33

- Points
- Lines
- Areas

# Points

34

- “A point represents a location on the plane that has no theoretical length or area. This signification is independent of the size and character of the mark which renders it visible.”
- a location
- marks that indicate points can vary in all visual variables

## Lines

35

- "A line signifies a phenomenon on the plane which has measurable length but no area. This signification is independent of the width and characteristics of the mark which renders it visible."
- a boundary, a route, a connection

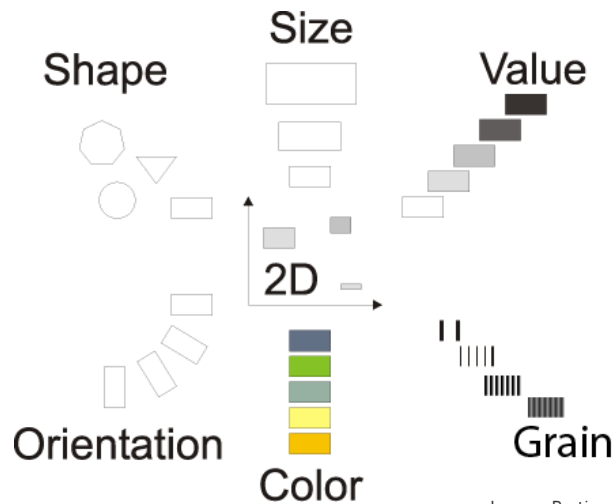
## Areas

36

- "An area signifies something on the plane that has measurable size. This signification applies to the entire area covered by the visible mark."
- an area can change in position but not in size, shape or orientation without making the area itself have a different meaning

# Visual Variables

37



# Additional Variables for Computers

38

- **motion**
  - ▣ direction, acceleration, speed, frequency, onset, 'personality'
- **saturation**
  - ▣ colour as Bertin uses it largely refers to hue, other readily available colour channels (*i.e.* saturation)
- **flicker**
  - ▣ frequency, rhythm, appearance
- **depth? 'quasi' 3D**
  - ▣ depth, occlusion, aerial perspective, binocular disparity
- **illumination**
- **transparency**

# Characteristics of Visual Variables

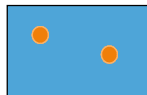
39

- **Selective:**  
Is a change in this variable enough to allow us to select it from a group?
- **Associative:**  
Is a change in this variable enough to allow us to perceive them as a group?
- **Quantitative:**  
Is there a numerical reading obtainable from changes in this variable?
- **Order:**  
Are changes in this variable perceived as ordered?
- **Length:**  
Across how many changes in this variable are distinctions possible?

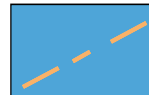
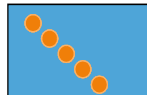
## Visual Variable: Position

40

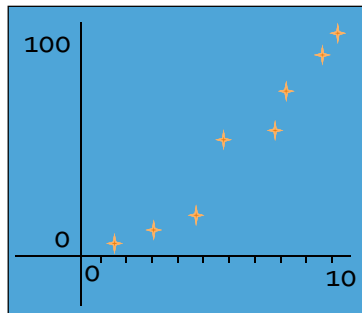
✓ □ selective



✓ □ associative



✓ □ quantitative



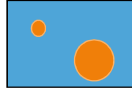
✓ □ order

✓ □ length

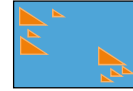
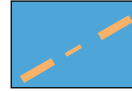
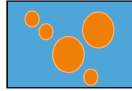
# Visual Variable: Size

41

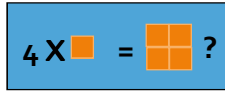
✓ □ selective



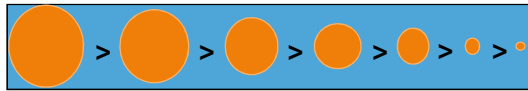
✓ □ associative



✗ □ quantitative



✓ □ order



✓ □ length

- theoretically infinite but practically limited
- association and selection ~ 5 and distinction ~ 20

# Size

42



points




lines

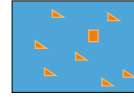
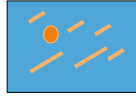


areas

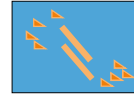
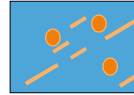
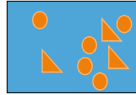
# Visual Variable: Shape


43

 □ selective



 □ associative



 □ quantitative



 □ order

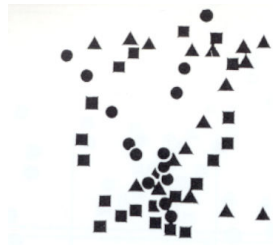
 □ length



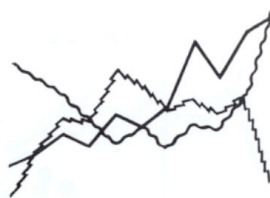
□ infinite

# Shape

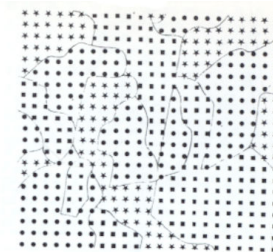
44



points



lines

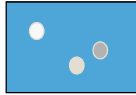


areas

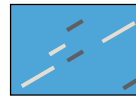
# Visual Variable: Value

45

✓ □ selective



✓ □ associative



✗ □ quantitative

✓ □ order

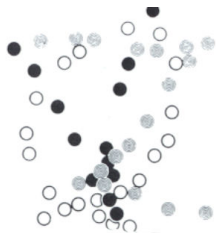


✓ □ length

- theoretically infinite but practically limited
- association and selection ~ < 7 and distinction ~ 10

# Value

46



points



lines

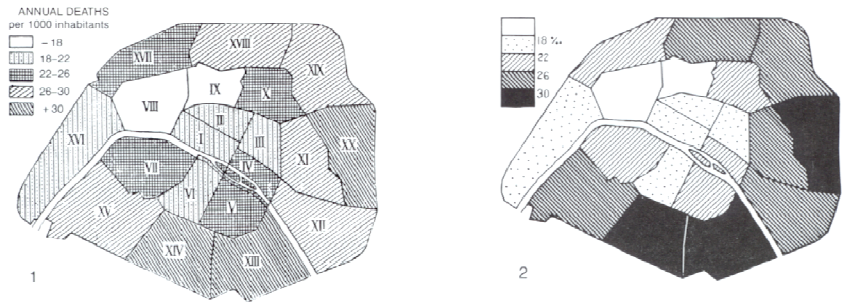


areas

# Value

47

- Ordered, cannot be reordered



annual deaths per 1000 inhabitants, Paris

# Visual Variable: Colour

48

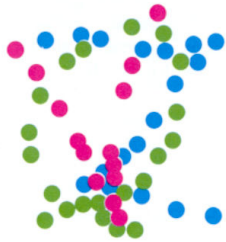
- selective
  - associative
  - quantitative
  - order
  - length
    - theoretically infinite but practically limited
    - association and selection ~ < 7 and distinction ~ 10





# Colour

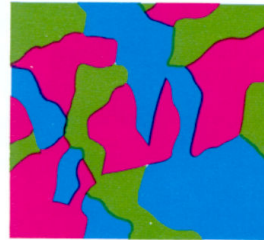
49



points



lines



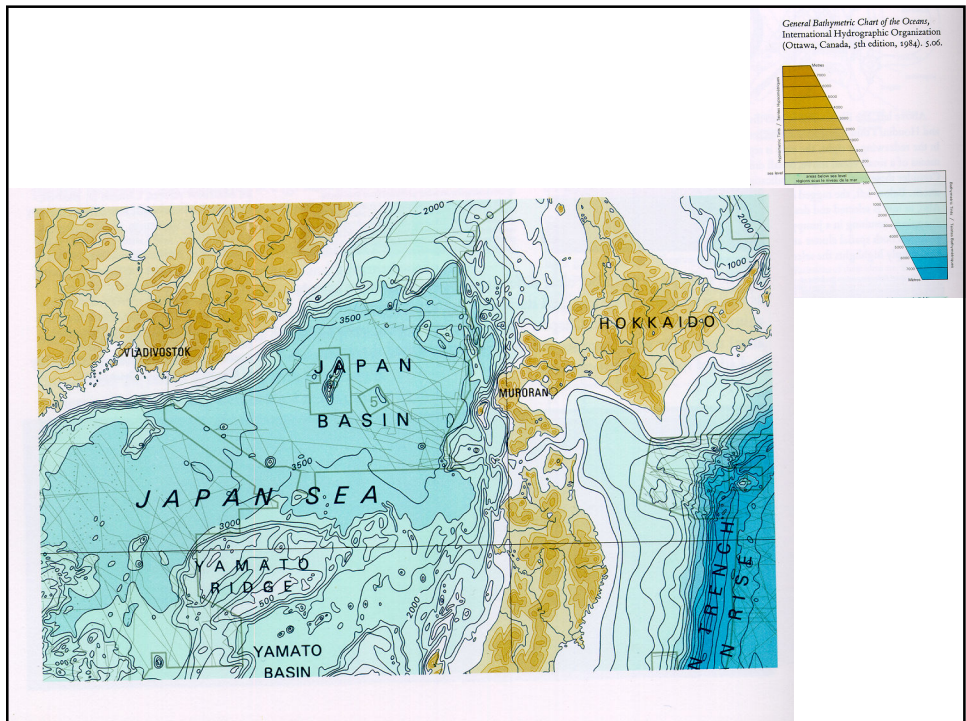
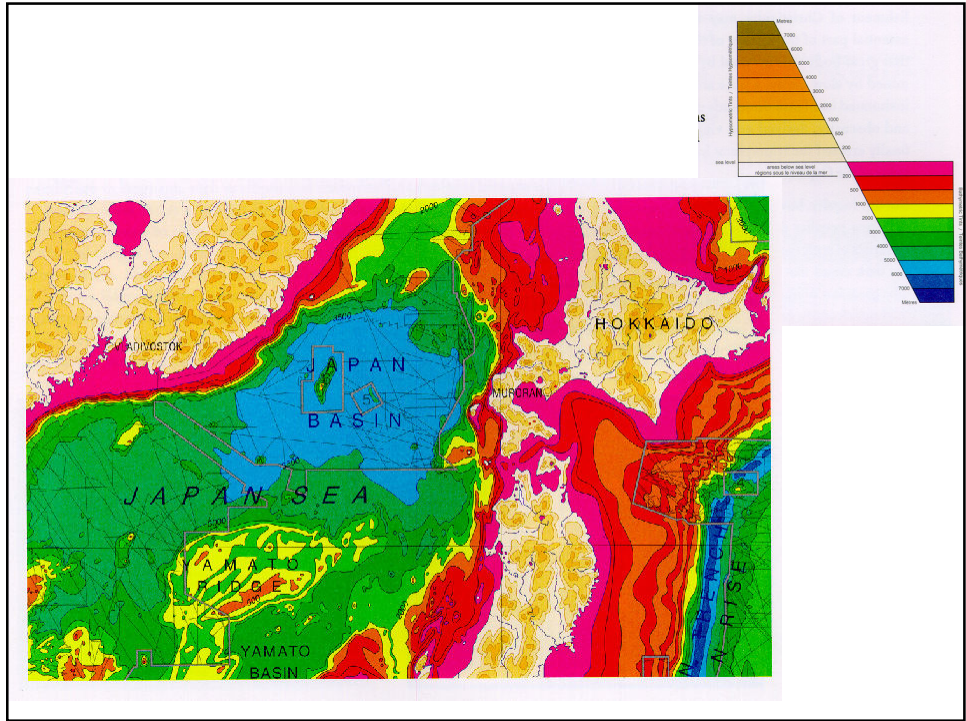
areas

# Colour Scales

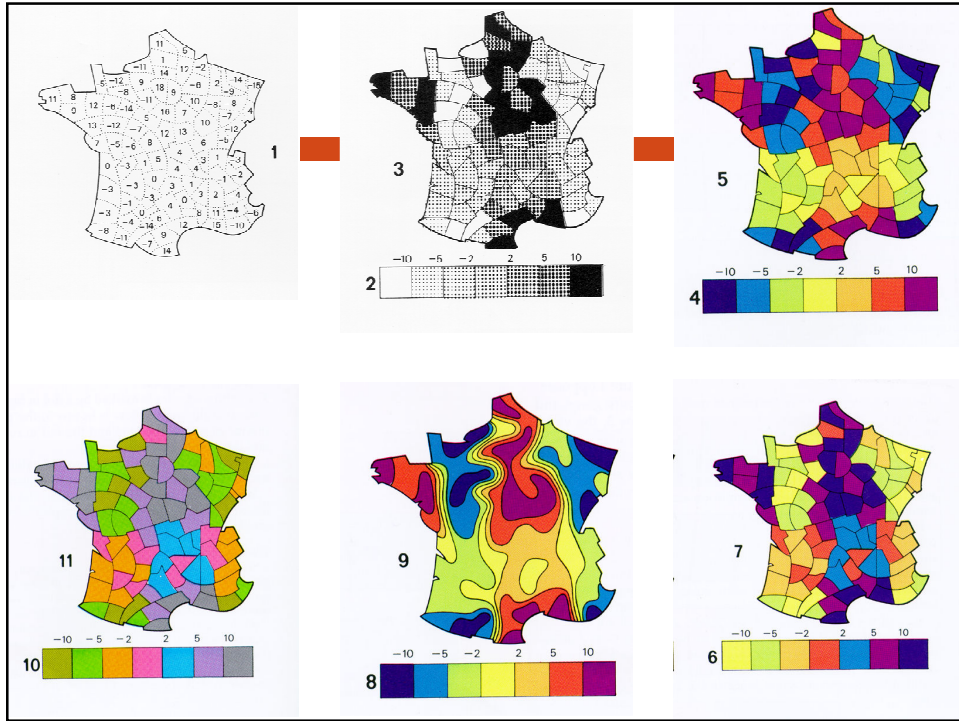
50

- Common to use a rainbow scale... caution!





General Bathymetric Chart of the Oceans, International Hydrographic Organization (Ottawa, Canada, 5th edition, 1984), 5.06.



## Controls Legibility

54

	R	G	B
Helvetica-plain/Helvetica-plain/Helvetica-plain/Helvetica-plain/Helvetica-plain/	0	0	0
Helvetica-plain/Helvetica-plain/Helvetica-plain/Helvetica-plain/Helvetica-plain/	0	31	0
Helvetica-plain/Helvetica-plain/Helvetica-plain/Helvetica-plain/Helvetica-plain/	0	63	0
Helvetica-plain/Helvetica-plain/Helvetica-plain/Helvetica-plain/Helvetica-plain/	0	95	0
Helvetica-plain/Helvetica-plain/Helvetica-plain/Helvetica-plain/Helvetica-plain/	0	127	0
Helvetica-plain/Helvetica-plain/Helvetica-plain/Helvetica-plain/Helvetica-plain/	0	159	0
Helvetica-plain/Helvetica-plain/Helvetica-plain/Helvetica-plain/Helvetica-plain/	0	191	0
Helvetica-plain/Helvetica-plain/Helvetica-plain/Helvetica-plain/Helvetica-plain/	0	223	0
Helvetica-plain/Helvetica-plain/Helvetica-plain/Helvetica-plain/Helvetica-plain/	0	255	0

255,255,255

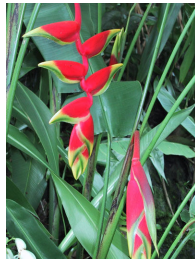
127,127,127

0,0,0

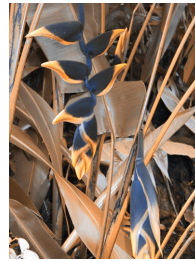
# Colour Blindness

55

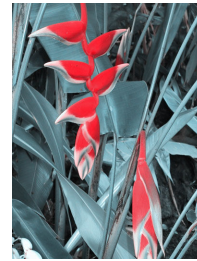
- VisCheck.com (Robert Dougherty and Alex Wade)
  - Web service to simulate colour blindness
- Biggest problems with red/green
- Varying value and hue can help



Deuteranope



Protanope

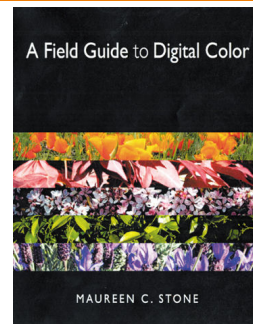


Tritanope

# Colour Resources

56

- Maureen Stone's Resources
  - <http://www.stonesc.com>
  - *A Field Guide to Digital Color* (A. K. Peters)
- Cindy Brewer's *ColorBrewer*
  - <http://colorbrewer.org>
- ColourLovers Community Palette Sharing
  - <http://www.colourlovers.com>



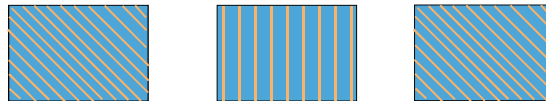
# Visual Variable: Orientation

57

✓ □ selective



✓ □ associative

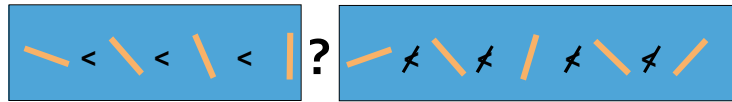


≠ □ quantitative

≠ □ order

✓ □ length

□ ~5 in 2D; ? in 3D



# Orientation

58



points



lines

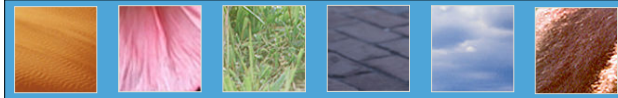


areas

# Visual Variable: Texture

59

✓ □ selective

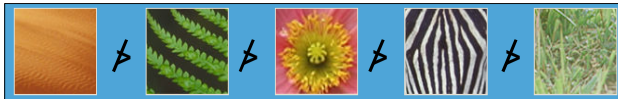


✓ □ associative



≠ □ quantitative

≠ □ order



✓ □ length

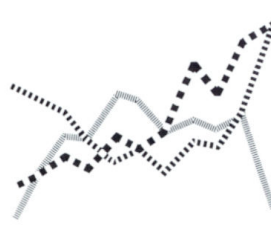
□ theoretically infinite

# Texture

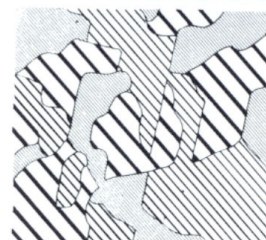
60



points



lines



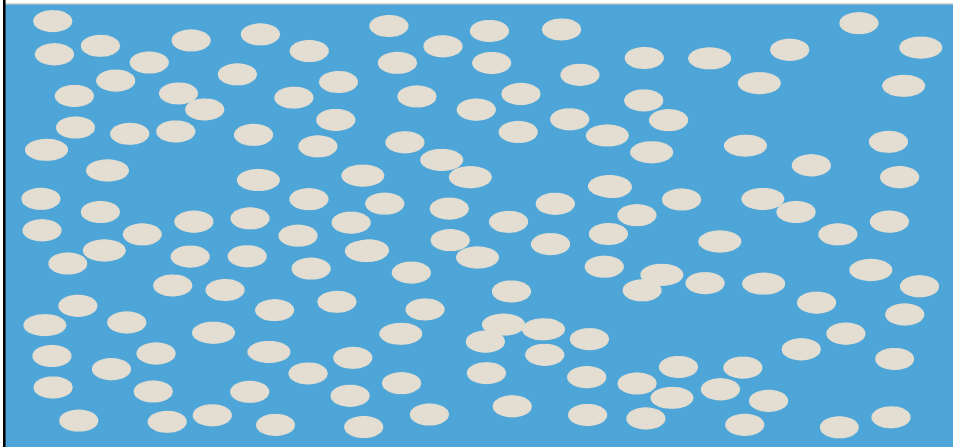
areas

# Visual Variable: Motion

- ✓ □ selective
  - ▣ motion is one of our most powerful attention grabbers
- ✓ □ associative
  - ▣ moving in unison groups objects effectively
- ≠ □ quantitative
  - ▣ subjective perception
- ≠ □ order
- ? □ length
  - ▣ distinguishable types of motion?

# Motion

62



# Visual Variables

63

Visual Variable	Selective	Associative	Quantitative	Order	Length
Position	Yes	Yes	Yes	Yes	Dependant on resolution
Size	Yes	Yes	Approximate	Yes	Association: 5; Distinction: 20
Shape	With Effort	With Effort	No	No	Infinite
Value	Yes	Yes	No	Yes	Association: 7; Distinction: 10
Hue	Yes	Yes	No	No	Association: 7; Distinction: 10
Orientation	Yes	Yes	No	No	4
Grain	Yes	Yes	No	No	5
Texture	Yes	Yes	No	No	Infinite
Motion	Yes	Yes	No	Yes	Unknown

Carpendale, 2003

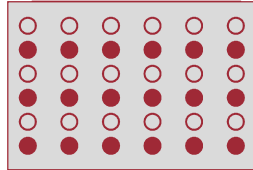
*Perception*

64

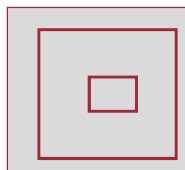


# Visual Gestalt

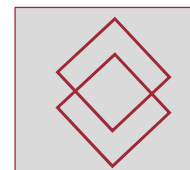
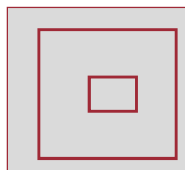
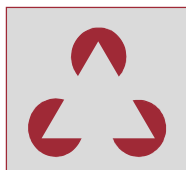
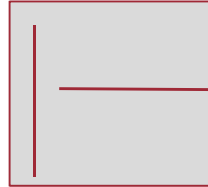
proximity



similarity



continuity



closure

figure/ground

symmetry

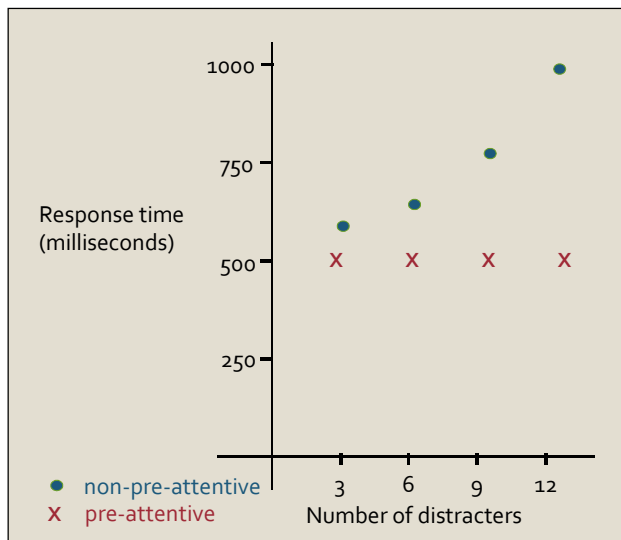
# Pre-attentive processing

66

2358945739756860796752453512346534624356245762457245  
6134523523523523523524351345324716498762987460329587  
2358276533637872138764298769876364098721696532962413  
9237462163987639871236597124593874638746988712649817  
2649872165971523972356987129721653978216409871246478  
3467218987639450897764398217346946496439276430987263  
4287469864987597152397123976490871469876498724369812  
7346987461435895321456865437

235894573975686**0**796752453512346534624356245762457245  
613452352352352352352435134532471649876298746**0**329587  
2358276533637872138764298769876364**0**98721696532962413  
9237462163987639871236597124593874638746988712649817  
26498721659715239723569871297216539782164**0**9871246478  
346721898763945**0**89776439821734694649643927643**0**987263  
428746986498759715239712397649**0**871469876498724369812  
7346987461435895321456865437

# Typical Results



# Pre-attentive processing

orientation



curved/straight



shape



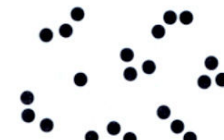
shape



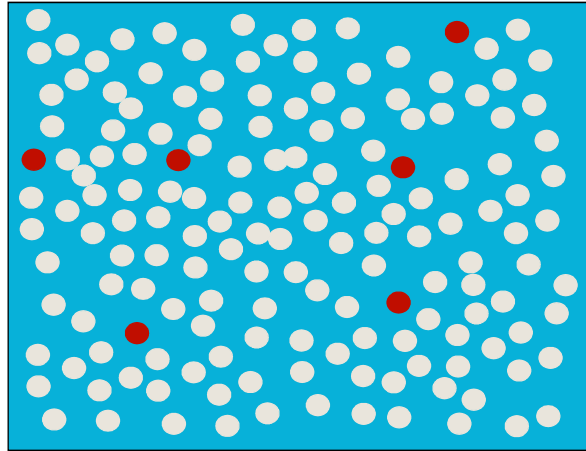
size



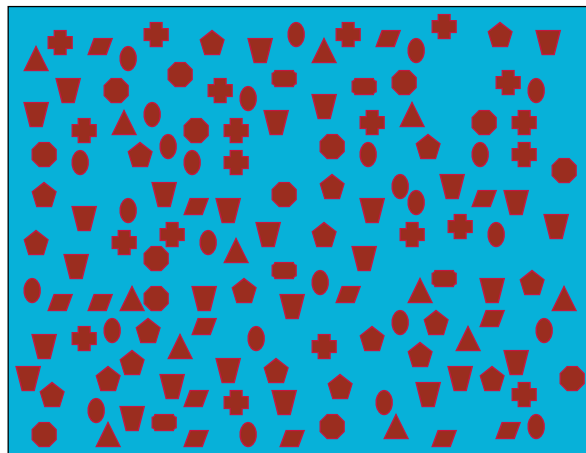
number



## Colour



## Shape



## *Interaction and Animation*

71

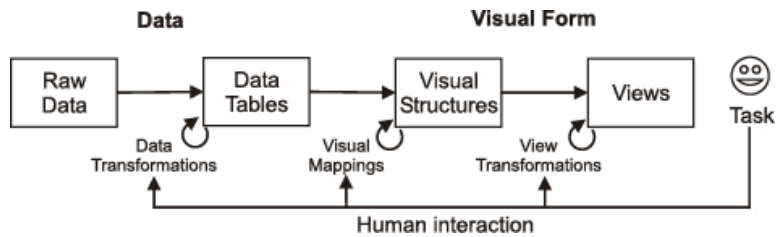
## Why Interaction?

72

- Datasets are too large to:
  - display in one view
  - comprehend in entirety
- Interest in only subset of the data
- Interest in different views of the data
- Extract relevant information & transform
- ...

# Sense-making Cycle

73



Card et al., 1999

# Interaction Techniques

74

Based on user intent

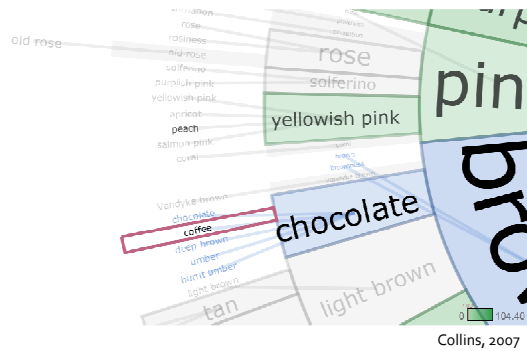
- Select – *mark something as interesting*
- Explore – *show me something else*
- Reconfigure – *show me a different arrangement*
- Encode – *show me a different representation*
- Abstract/Elaborate – *more or less detail*
- Filter – *show me something conditionally*
- Connect – *show me related items*

Yi et al., InfoVis 2007

# Selection

75

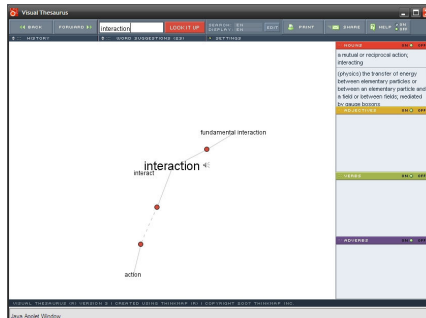
- Mark something as interesting
- Often combined with other techniques



# Explore

76

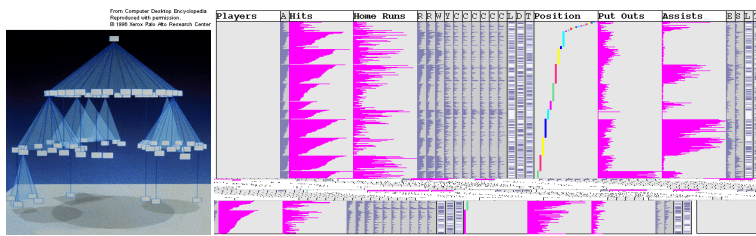
- Show me something else
- Examine subset of data cases (view-based)
  - E.g. Panning (move viewpoint across representation)
  - E.g. Direct Walk (move viewing focus through clicks)



# Reconfigure

77

- Show a different arrangement
  - ▣ Move data items to
    - Enable better comparison
    - Avoid occlusion
    - Correspond to some mental model of the data



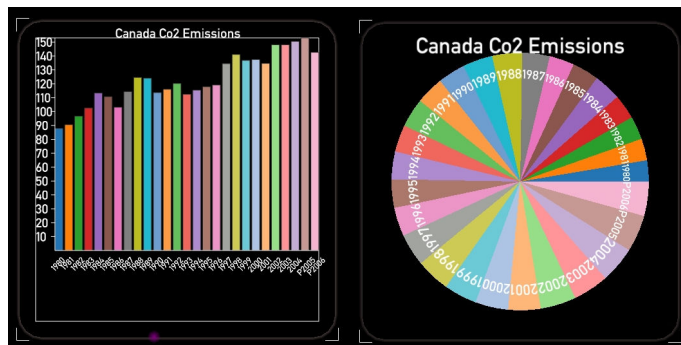
Cone Trees  
(Robertson et al., 1991)

Table Lens  
(Rao & Card, 1995)

# Encode

78

- Show a different:
  - ▣ Representation Type
  - ▣ Visual appearance: Colour, Size, Shape,...

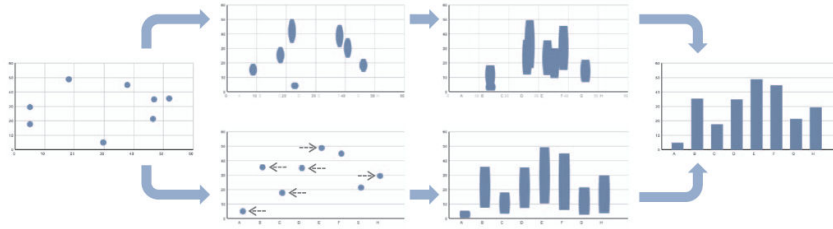


Isenberg and Carpendale, InfoVis, 2007

# Encode

79

- Animation can aid encoding changes

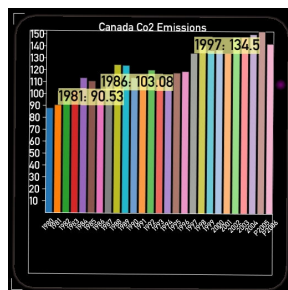


DynaVis - Heer & Robertson, 2007

# Abstract/Elaborate

80

- Show me more or less detail
  - Adjust level of abstraction
  - Detail-on-demand
  - Zooming (as long as representation isn't fundamentally altered)



Isenberg and Carpendale, InfoVis, 2007

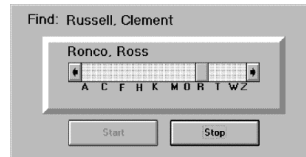
**Warning:** Not every technique belongs to just one category.



# Filter

81

- Show subset of data based on condition
  - ▣ e.g., by selecting a data range



- ▣ or filtering based on distance from focus

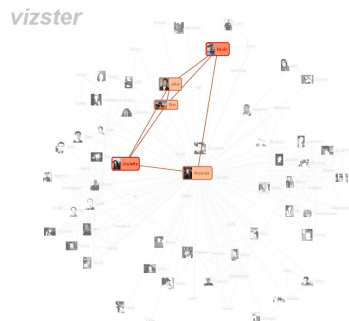


AlphaSlider, Ahlberg & Shneiderman, 1993; DocuBurst, Collins, 2007

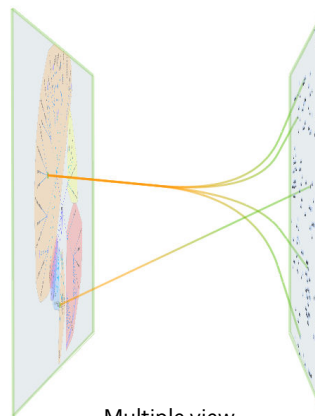
# Connect

82

- Show related items
  - ▣ e.g. brushing

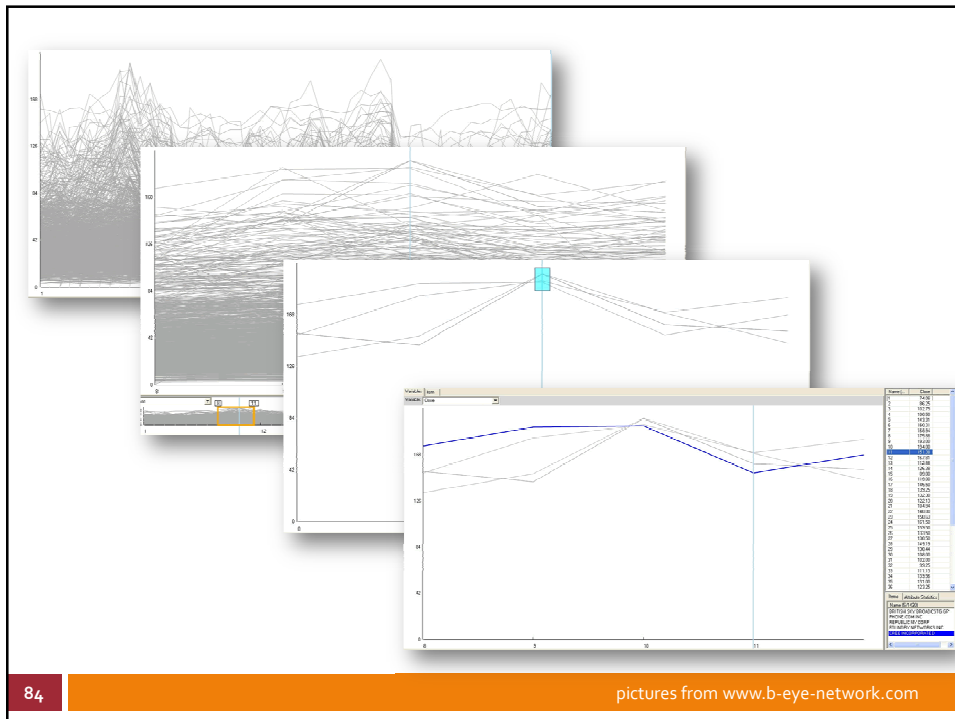


Single view  
Heer & boyd, InfoVis 2005



Multiple view  
Collins & Carpendale, InfoVis 2007

*"Overview first, zoom and filter,  
then details-on-demand"*



## *Assessing and Validating*

85

## Evaluation Challenges

86

- Evaluations not convincing enough
  - ▣ Small datasets, unrealistic participant samples, simple tasks
- Common challenges
  - ▣ Right focus
  - ▣ Right questions
  - ▣ Right methodology
  - ▣ Proper procedure
  - ▣ Correct stats
  - ▣ Cross validation
  - ▣ Relating results to the field
- Entire conferences dedicated to InfoVis Evaluation!

# Types of Evaluation

87

- Proofs
  - ▣ Verify the mapping; prove correctness
  - ▣ Show algorithms are faster, improved complexity
- Usability
  - ▣ Empirical usability evaluation of system
  - ▣ Perceptual study on a small part
- Case Studies
  - ▣ Show that you can see things previously hidden
  - ▣ Longitudinal case study
  - ▣ Insight study

# Longitudinal Case Study

88

- Deploy visualization to data domain experts
  - ▣ Do they use it?
  - ▣ Do they like it?
  - ▣ Does it save them time?
  - ▣ Are there indirect benefits? Do their results (e.g. BLEU score, parser accuracy) improve?
  - ▣ Improved collaboration?
- Techniques include interviews, questionnaires, in situ observation, and automated logging

## Measuring Insight

89

- Controlled experiment with data domain experts
  - ▣ Domain professionals carry out controlled analysis and record observations, conclusions, insights
  - ▣ Experts in the specific dataset reviews and rates insights for originality, correctness, importance
- Metrics: Insights/time, Total insights, Correctness
- Difficult to recruit sufficient quantities of experts
- Simultaneously study other measures: usability issues, visualization techniques used, ...

Saraiya et al., IEEE TVCG 11(4), pp. 443-457, 2005

## Heuristic Approaches

90

- Heuristics can be useful if learned and internalized
  - ▣ "generally avoid red-green colour scales"
  - ▣ "don't use shape for quantitative encoding"
- Careful! There are no "rules"!
- Heuristic evaluation can lead to lowest common denominator designs!

# Visualization Design

91

## Data Processing is Task Dependent

92

- What are the information needs?
- What errors may be introduced by processing?

El<sup>1</sup> español<sup>2</sup> es<sup>3</sup> la<sup>4</sup> lengua<sup>5</sup> más<sup>6</sup> hablada<sup>7</sup> del<sup>8</sup> mundo<sup>9</sup> tras<sup>10</sup> el<sup>11</sup> chino<sup>12</sup> mandarín<sup>13</sup> por<sup>14</sup> el<sup>15</sup> número<sup>16</sup> de<sup>17</sup> hablantes<sup>18</sup> que<sup>19</sup> la<sup>20</sup> tienen<sup>21</sup> como<sup>22</sup> lengua<sup>23</sup> materna<sup>24</sup>.

Spanish<sup>1,2</sup> (0.90) is<sup>3</sup> (0.90) the<sup>4</sup> (0.94) language<sup>5</sup> (0.39) most<sup>6</sup> (0.25) spoken<sup>7</sup> (0.52) [about the]<sup>8</sup> (0.30) world<sup>9</sup> (0.93) following<sup>10</sup> (0.64) the<sup>11</sup> (0.72) Chinese<sup>12</sup> (0.87) Mandarin<sup>13</sup> (0.80) for<sup>14</sup> (0.24) the<sup>15</sup> (0.77) number<sup>16</sup> (0.79) of<sup>17</sup> (0.99) speakers<sup>18</sup> (0.82) who<sup>19</sup> (0.80) take<sup>21</sup> (0.21) it<sup>20</sup> (0.96) [as a]<sup>22</sup> (0.46) mother<sup>24</sup> (0.35) tongue<sup>23</sup> (0.25)

Spanish<sup>1,2</sup> (0.90) is<sup>3</sup> (0.90) the<sup>4</sup> (0.83) most<sup>6</sup> (0.55) spoken<sup>7</sup> (0.73) language<sup>5</sup> (0.44) [in the]<sup>8</sup> (0.89) world<sup>9</sup> (0.94) after<sup>10</sup> (0.73) Chinese<sup>11,12</sup> (0.80) Mandarin<sup>13</sup> (0.88) by<sup>14</sup> (0.41) the<sup>15</sup> (0.79) number<sup>16</sup> (0.94) of<sup>17</sup> (0.75) speakers<sup>18</sup> (0.84) that<sup>19</sup> (0.62) have<sup>21</sup> (0.22) as<sup>22</sup> (0.40) their<sup>20</sup> (0.10) mother<sup>24</sup> (0.37) tongue<sup>23</sup> (0.18).

....

# Graphical Excellence

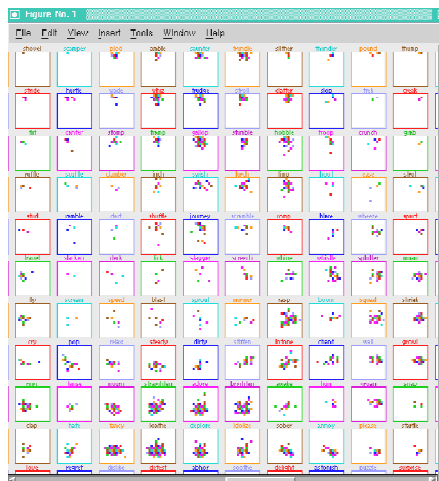
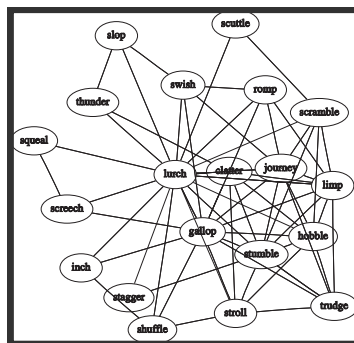
93

- Principles of **graphical excellence**
  - ▣ Complex ideas presented with clarity, precision, honesty, and efficiency.
  - ▣ Gives viewer the greatest number of **ideas** in the shortest **time**, with the least **ink** in the smallest **space**.
- Graphical excellence often found in **simplicity of design** and **complexity of data**.
  - ▣ multivariate data, simple design

Tufte, 2001

# Reduce Learning Curve: Small Multiples

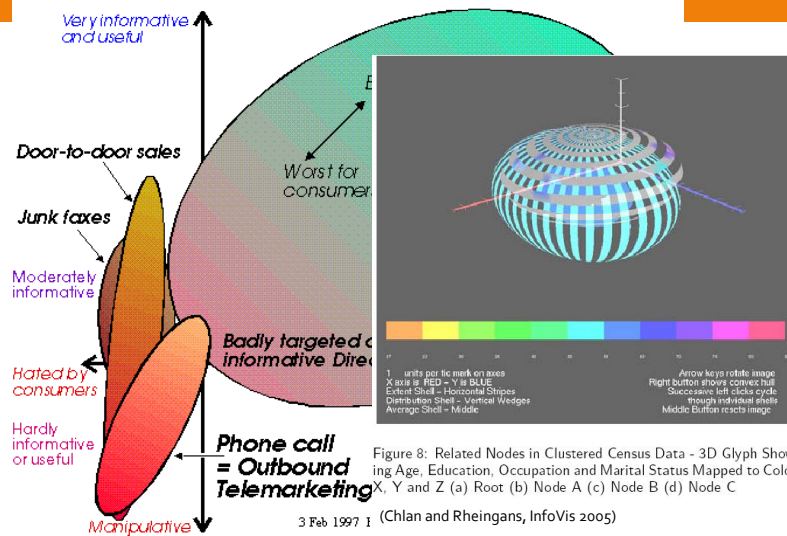
94



N. Theissen, 2004

## High Dimensionality Doesn't Guarantee Excellence

95



## Visualization Pitfalls

96

- **Data Design**
  - ▣ Select the right data dimensions
  - ▣ **Pitfall:** Display irrelevant data relationships
- **Visual Design**
  - ▣ Consider perceptual capabilities
  - ▣ **Pitfall:** Difficult to interpret, lead users to misinterpretation
- **Interface Design**
  - ▣ Mode of interaction with visual and data appropriate
  - ▣ **Pitfall:** Poor interaction negates benefits from data and visual design
- **Understanding Target Users**
  - ▣ Visualization design accounts for stakeholder needs and characteristics
  - ▣ **Pitfall:** Mistaken assumptions of cultural norms or user abilities leads to misinterpretation

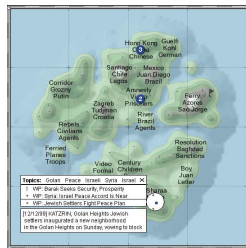
Collins



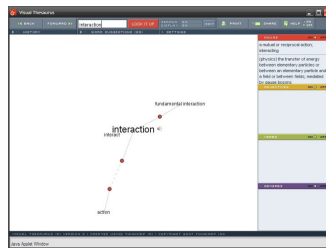
# Externalizing Language

97

- To assist information retrieval
- To enable linguistic analysis
- To augment analytics on mixed data



Themescape (Wise, 1995)



Visual Thesaurus



Thread Arcs (Kerr, 2003)

98

## Review of Linguistic Visualizations

## 'Language' Data

- Text
  - ▣ Content, form, temporality, repetition, semantics, origin
- Speech
  - ▣ Content, tonality, prosody, temporality, semantics, origin
- Other forms of language:
  - ▣ Sign languages
  - ▣ Sets of commonly known symbols (semiotics)
  - ▣ Mixed data (e.g., text + social networks, music)
  - ▣ Multimodal communications (gesture + speech)

## Difficult Data

100

- Too much data – what to use?
  - ▣ Millions of blog posts,
  - ▣ Hundreds of thousands of news stories,
  - ▣ 183 billion emails,
  - ▣ ... **per day**
  - ▣ + all the LDC corpora!
- Data is noisy:
  - ▣ Newswire stories are syndicated (but differ slightly)
  - ▣ 70-72% of email is spam
  - ▣ Text contains section headings, figure captions, and direct quotes
  - ▣ Annotator disagreements

## Data Processing Decisions

101

- Many levels of data processing can take place:
  - ▣ Word counting
  - ▣ Stemming: "reads" and "reading" → "read"
  - ▣ Parsing: "USA invaded Iraq" → Invaded(USA,IRAQ)
  - ▣ Summarization
  - ▣ Sentiment analysis: "Electronics shops have terrible customer service" → negative assessment
  - ▣ Word sense disambiguation: "We go to the bank to obtain a loan to purchase a boat" → bank=financial institution(72%)
- Each step of extra processing introduces uncertainty and takes time

## Language is Ambiguous

102

- Words and phrases can have many meanings, determined by context and world knowledge.
- Ambiguities abound in CL research, e.g.:
  - ▣ PP attachment
  - ▣ Word sense
  - ▣ Lexical transfer/alignment in MT
  - ▣ Idiomatic/literal interpretations
  - ▣ ...

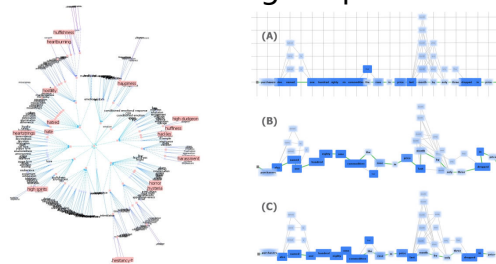
## Visualizing Language is Difficult

- Many of the common challenges still exist:
  - Screen real estate / occlusion
  - Choosing appropriate visual variable mappings
  - Colour and perception issues
  - Maintaining “graphical integrity”
  - Interaction and usability
- Specific challenges for language?

## Labelling

104

- A general issue in information visualization
- How to label a visualization while:
  - Positioning labels in a meaningful way
  - Making them long enough to be useful
  - Avoiding label overlap
  - Fonts large enough to read; minimal rotation
- If data is textual, labelling is a problem.



## Visual Considerations

105

- 
- Text readability is dependent on size, orientation, font, clutter...

## Visual Considerations

106

Supporters of Martin, who has been jailed without trial for more than two years, are calling on Prime Minister Stephen Harper to ask Mexican president Felipe Calderon to release Martin text is not preattentive under a section of the Mexican constitution that allows the government to expel undesirables from the country. Martin's supporters believe she has no chance of a fair trial in Mexico. Neither does Waage.

## Pre-attentive processing

107

Supporters of Martin, who has been jailed without trial for more than two years, are calling on Prime Minister Stephen Harper to ask Mexican president Felipe Calderon to release Martin **text is not preattentive** under a section of the Mexican constitution that allows the government to expel undesirables from the country. Martin's supporters believe she has no chance of a fair trial in Mexico. Neither does Waage.

## Visualizing language is easy, right?

108

- SO much data available for analysis
- (Mostly) readily computer readable
- Simple techniques can give instant summaries
- *You* are the data processing experts!

god	3370	lord	7872
we	1971	god	4690
You	1544	me	4096
lord	955	son	3464
bath	851		

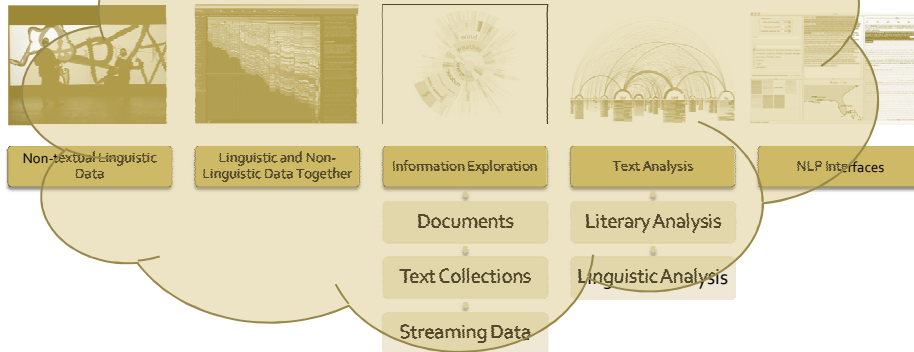
Simple techniques, though easy, are often insufficient for real analysis and discovery.

Most examples we will now show just use simple techniques... but they are good to know!

do	870	man	2013
make	549	king	2600
verily	479	house	2160
me	460	people	2139
send	437	child	2003
people	432	give	1872
earth	420	we	1844

# Visualization Categories

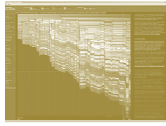
Stronger relationship to NLP research  
Greater sophistication of data processing



# Non-textual Linguistic Data



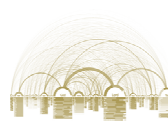
Non-textual Linguistic Data



Linguistic and Non-Linguistic Data Together



Information Exploration



Text Analysis



NLP Interfaces

Documents

Literary Analysis

Text Collections

Linguistic Analysis

Streaming Data





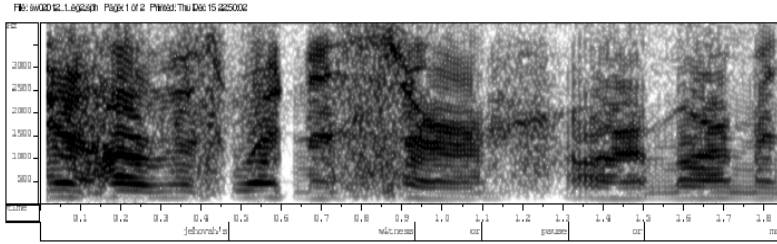
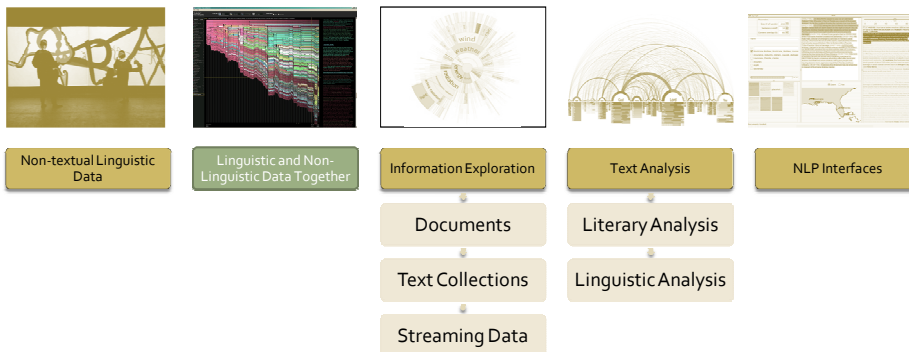


Figure 1: The pause between two *or* s and the glottalization at the end of the first makes it easy for a listener to identify the repair.

## Linguistic and Non-linguistic Data Together



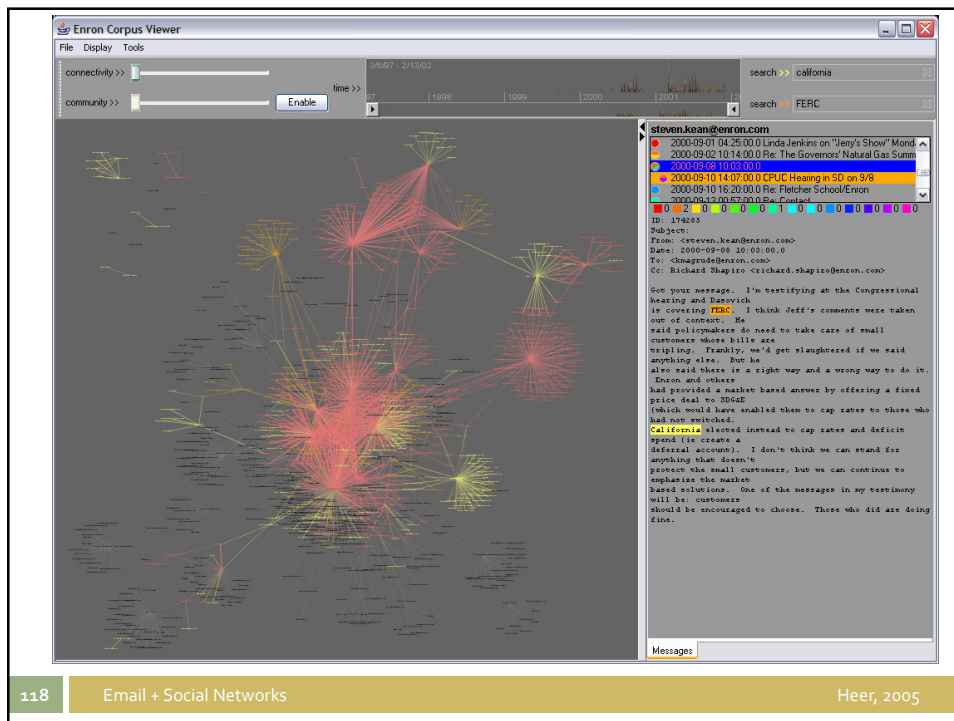




117

Language as Structural Scaffold

ColorCode, Wattenberg, 2005



118


Email + Social Networks

Heer, 2005

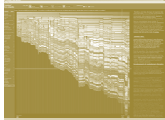
The screenshot displays a complex data analysis software interface. It features a central grid with columns numbered 1-14 and rows representing dates from 1900.02.18 to 1900.10.27. To the right is a map showing geographical locations. Below the grid are several data tables, including 'Visits', 'Residences', and 'Occupations'. The interface includes a menu bar at the top and various control panels for parameters and options.

119 Intelligence Analysis Hotel Records, Weaver et al., Proceedings of VAST, 2007


# Documents



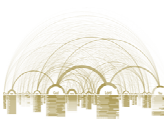
Non-textual Linguistic Data




Linguistic and Non-Linguistic Data Together



Information Exploration



Text Analysis



NLP Interfaces

Documents

Text Collections

Streaming Data

Literary Analysis

Linguistic Analysis

Information Exploration

120

*Alice's Adventures In Wonderland*

Text

Show only KMIC Index (Key Word in Context)

Show

Hide KMIC Project Gutenberg header

Download this e-book

Read

ALICE'S ADVENTURES IN WONDERLAND

Lewis Carroll

THE MILLENNIUM FULCRUM EDITION 3.0

CHAPTER I

Down the Rabbit-Hole

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, "and what is the use of a book," thought Alice "without pictures or conversation?"

So she was considering in her own mind (as well as she could, for the hot day made her feel very sleepy and stupid), whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a White Rabbit with pink eyes ran close by her.

There was nothing so VERY remarkable in that; nor did Alice think it so VERY much out of the way to hear the Rabbit say to itself, "Oh dear! Oh dear! I shall be later" (when she thought it over afterwards, it occurred to her that she ought to have wondered at this, but at the time it all seemed quite natural); but when the Rabbit actually TOOK A WATCH OUT OF ITS WAISTCOAT-

121

Using Document Structure

TextArc: Paley, InfoVis Poster 2002

artificial

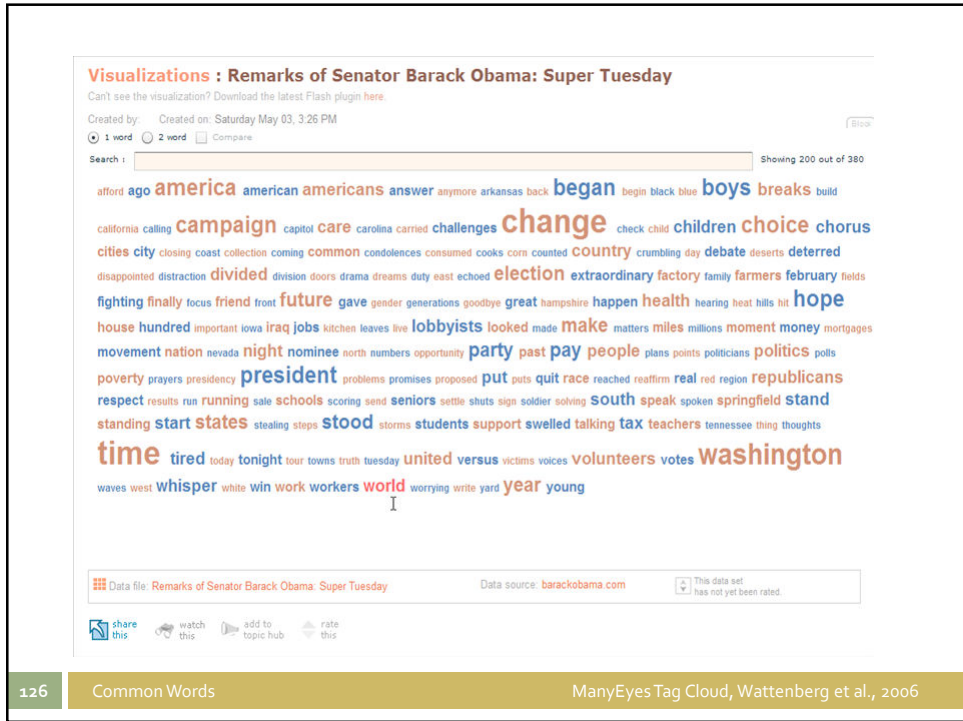
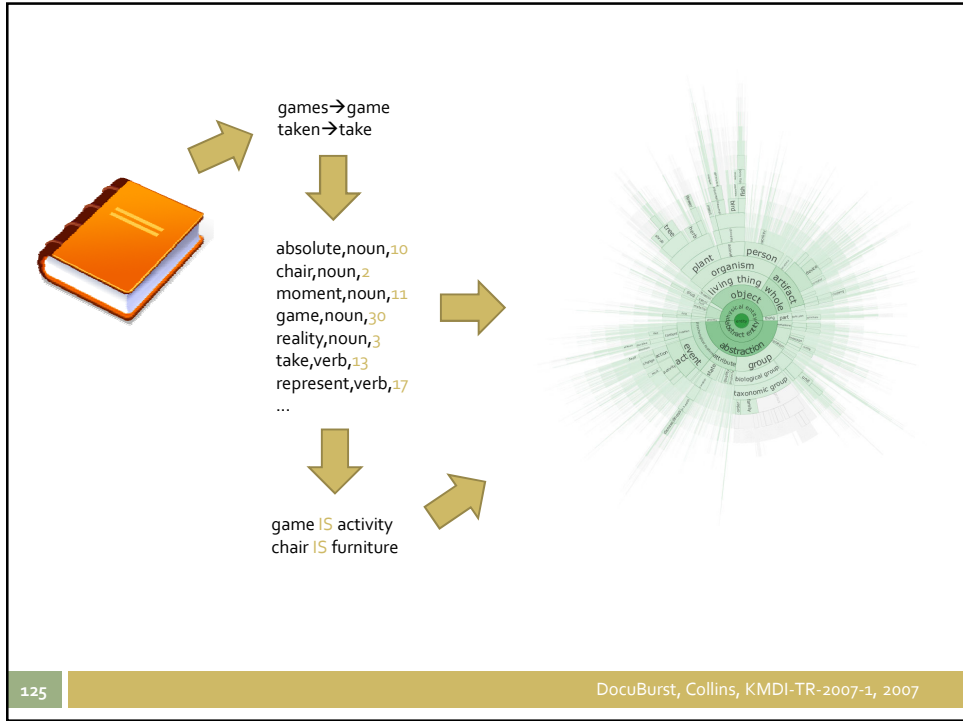
nature

122

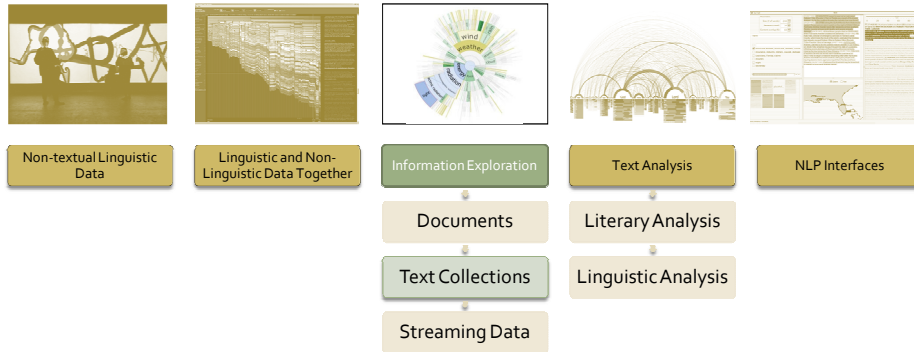
Using Statistical Analysis

Gist Icons: DeCamp et al., InfoVis Poster 2005



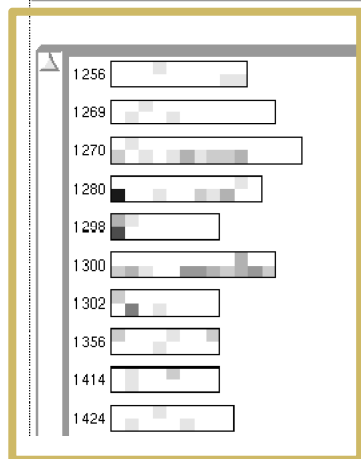


# Text Collections



Term Set 1: law legal attorney lawsuit

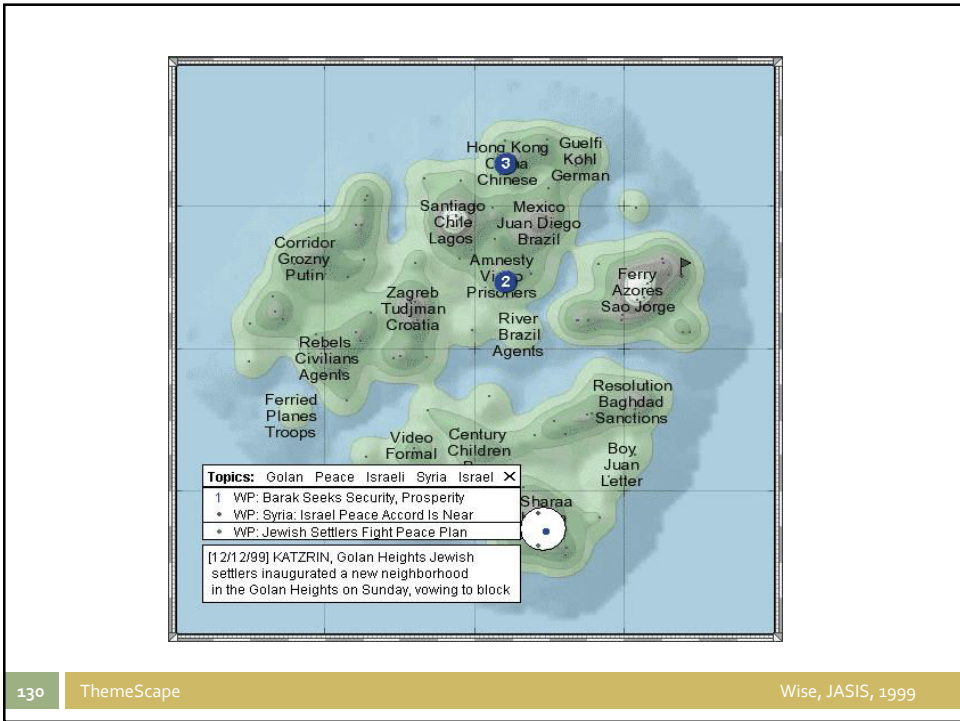
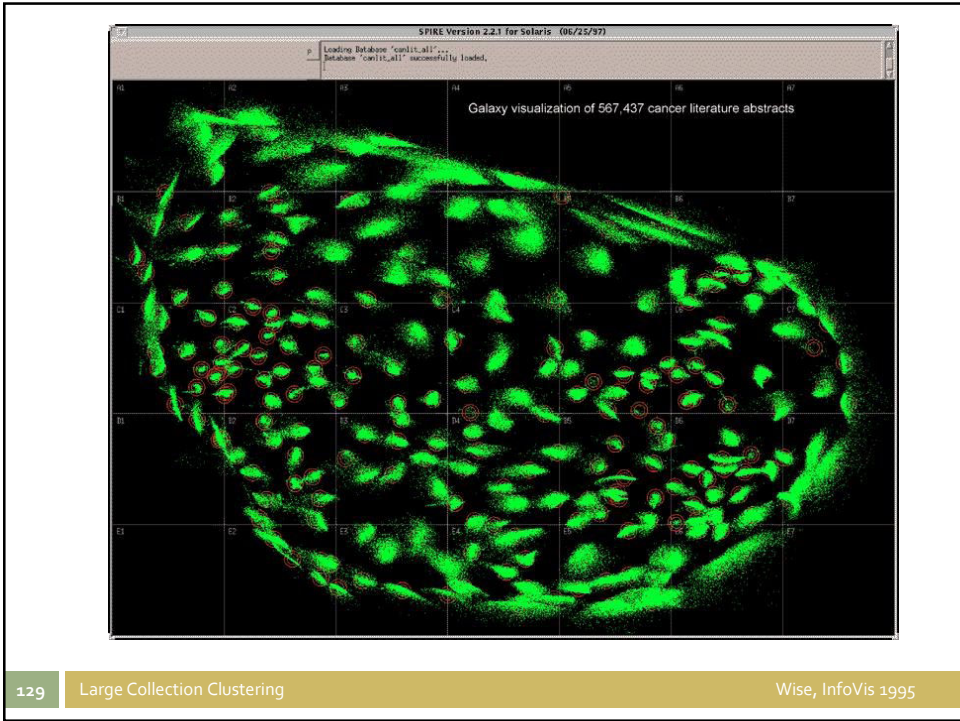
Term Set 2: network lan



## TileBars

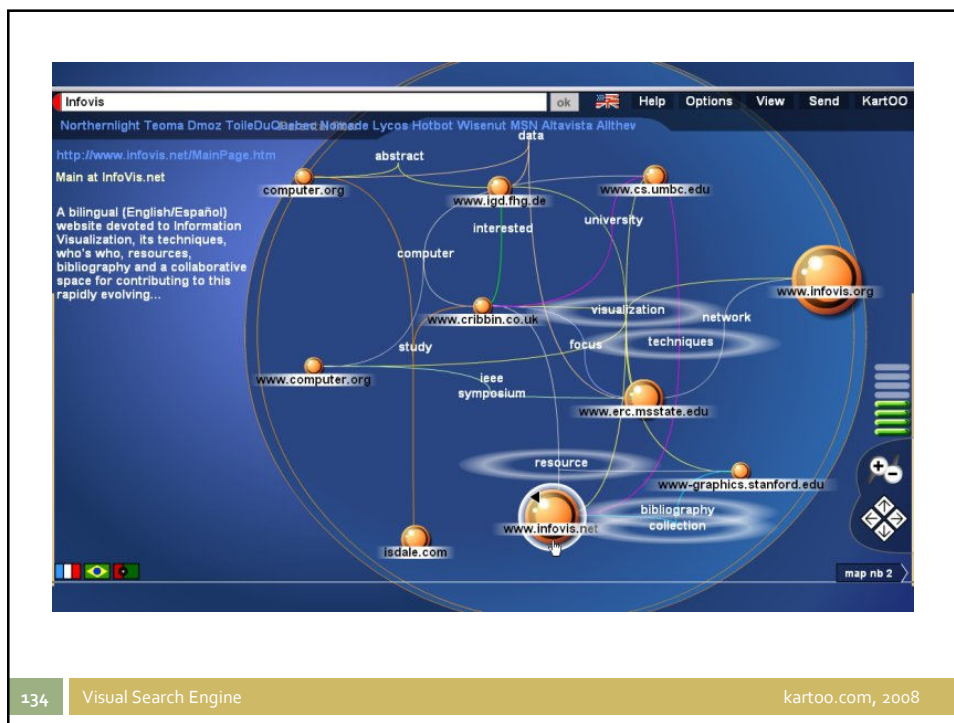
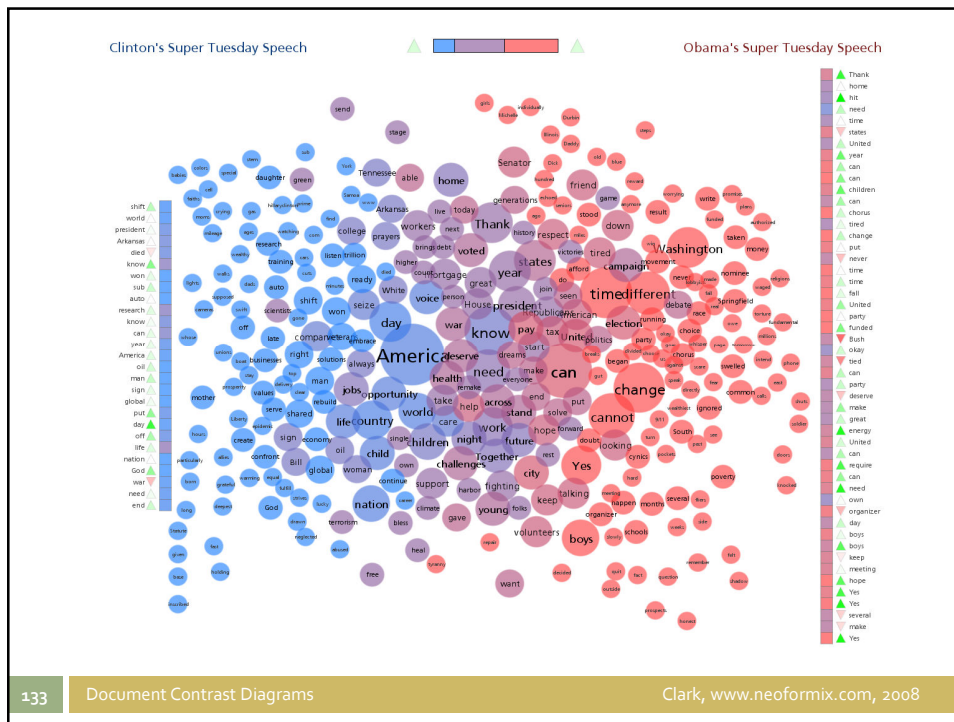
Regression testing handling ha  
 Toll fraud includes related articl  
 In conversation Teleglobe Can:  
 Deregulation indicates a health  
 The last word letters to the edi  
 What's wrong with network lice  
 Letters letter to the editor  
 Protecting information now vitz  
 Letters O  
 Loose LIPS sink ships logical ir



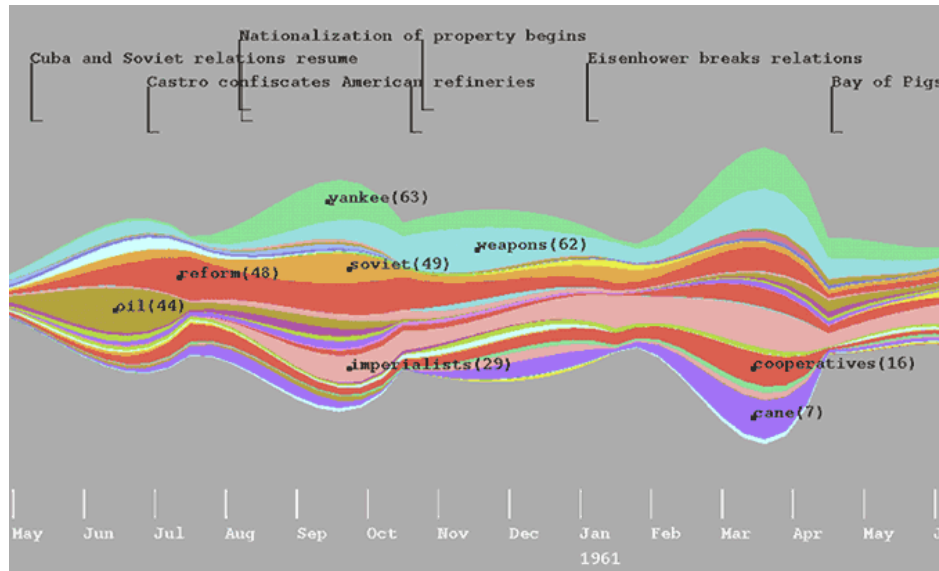
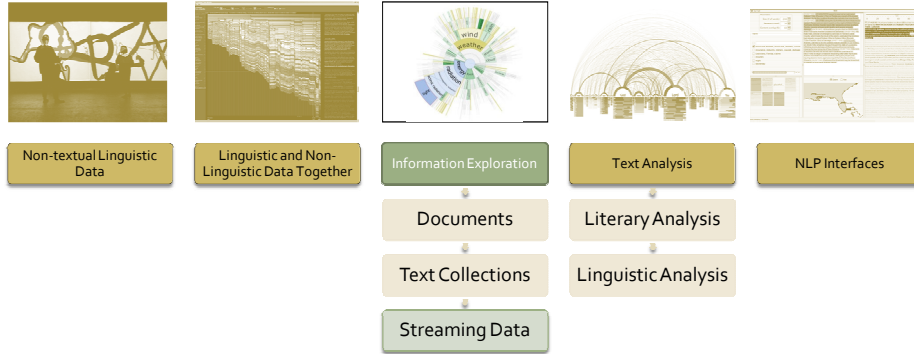


131 Cross-lingual IR Leuski et al., ACM Trans. on Asian Language Information Processing, 2003

132 Document Structure and Content Comparison Rembold and Späth, Total Interaction, 2006



# Streaming Data



Monday May 5, 2008 16:29

SELECT ALL CATEGORIES

WORLD BUSINESS TECHNOLOGY SPORTS ENTERTAINMENT HEALTH

LESS THAN 10 MINUTES OLD MORE THAN 10 MINUTES OLD MORE THAN 1 HOUR OLD

137 Streaming News, Thematic Weskamp, <http://marumushi.com/apps/newsmap/>, 2004

WE FEEL FINE

i feel like all i ever do is try to make him proud but i always seem to fail miserably

18 mins ago | from a female

Like 1

Loading We Feel Fine applet (8246)

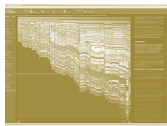
138 Streaming Blog Data, Emotional Focus Harris, [www.wefeelfine.org](http://www.wefeelfine.org), 2006



# Literary Analysis



Non-textual Linguistic Data



Linguistic and Non-Linguistic Data Together



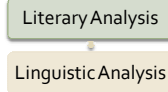
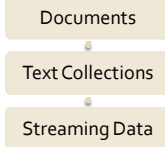
Information Exploration

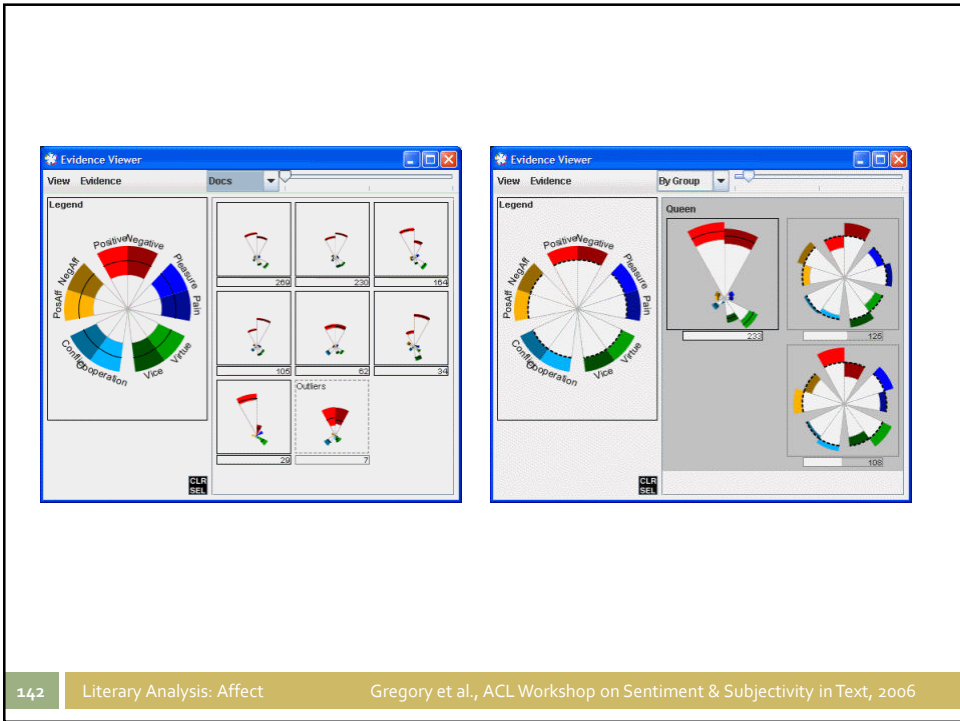
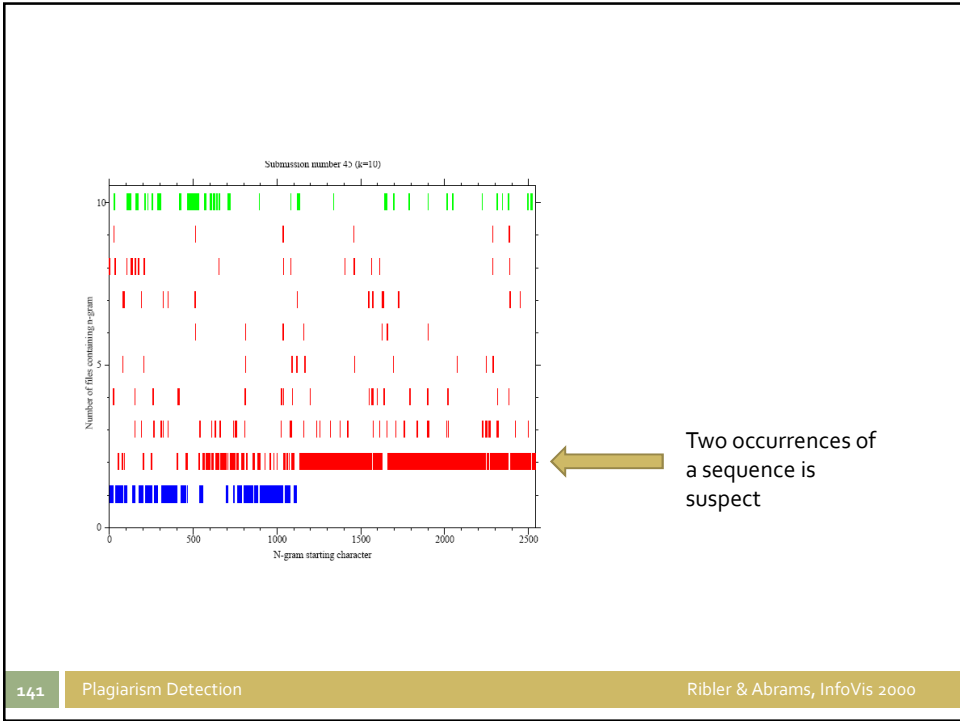


Text Analysis



NLP Interfaces

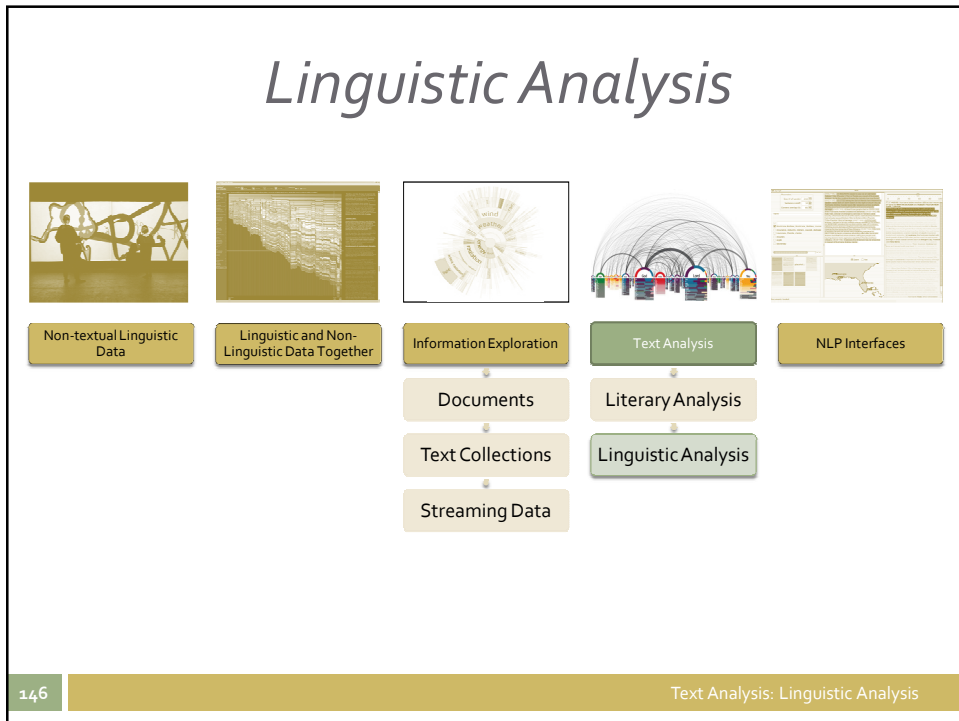




143 Literary Analysis: Patterns Feature Lens, Don et al., CIKM 2007

144 Literary Analysis: Repetition ManyEyes WordTree, Wattenberg et al., 2007





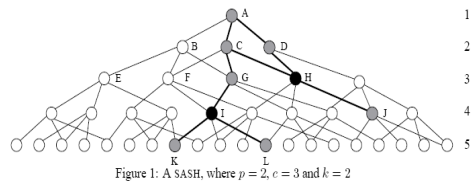
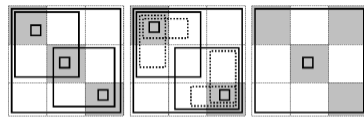


Figure 1: A SASH, where  $p = 2$ ,  $c = 3$  and  $k = 2$

Gorman and Curran, Scaling Distributional Similarity to Large Corpora



(a) Manual (b) Automated\_1 (c) Automated\_2

Figure 1: Sample phrases that are generated from a human alignment and an automated alignment. Gray cells show the alignment links, and rectangles show the possible phrases.  
Ayan and Dorr, Going Beyond AER: An Extensive Analysis of Word Alignments and Their Impact on MT

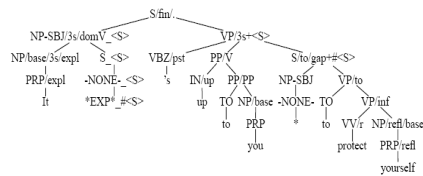
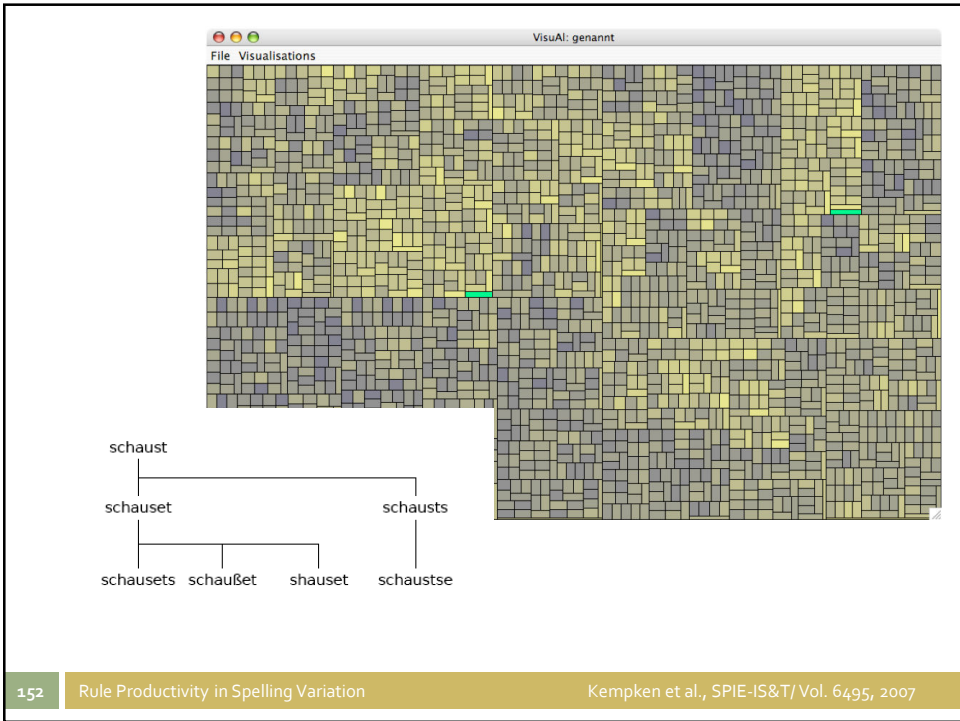
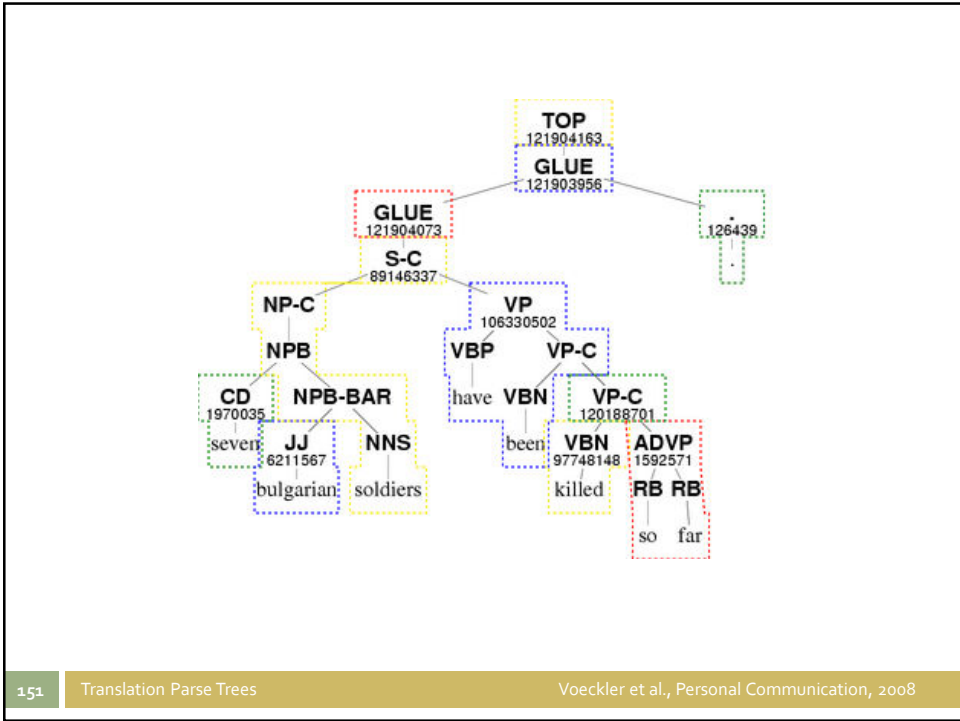
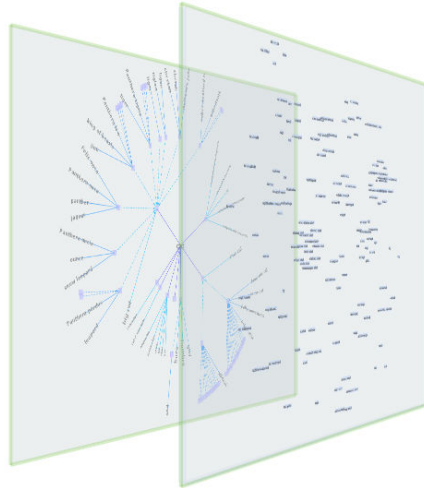


Figure 5: An Annotated Parse Tree

Schmid, Trace Prediction and Recovery with Unlexicalized PCFGs and Slash Features



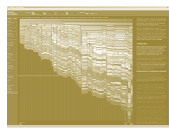




# NLP Interfaces



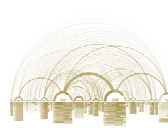
Non-textual Linguistic Data



Linguistic and Non-Linguistic Data Together



Information Exploration



Text Analysis



NLP Interfaces

Documents

Literary Analysis

Text Collections

Linguistic Analysis

Streaming Data

Parameters

Size (# of words): 200

Sentence cutoff: 10

Content overlap (%): 60

Topics

- Hurricane Andrew, Hurricane, Andrew, caused
- insurance, industry, Dollars, caused, damage
- Louisiana, Florida, claims
- insurers
- night
- yesterday

Documents loaded.

Zoom Pan

155 Multi-document summarization Leuski et al., ACL Demos, 2003

DerivTool 0.6

New Corpus Corpus=1-test.50 (60 lines)20

Go to line Load Save Print Print to File Redraw Close

Derivation Tree Freeform Notes

English: the gunman was killed by the police.

Click, Ctrl-Click, or drag above to select. For adding rules, please select a contiguous span of top level nodes.

Phrase-based MT: gunman killed by police

Searching Manual Selecting Template

Lines = 100 Redraw Click below to add rule. Red=no rules found, Blue=used by AT translation, Green=default translation, Purple-Red+Blue

gunmen	being	police	killed
by gunmen	were	traffic police	kill
gunman	been	police officers	ypogostav guards responded and killed
assaultants	are being	police force	killing
shooters	had been	police are	
	have been	police have	
	has been	a police	
		the police	
	by	police	

no rules are available for this phrase (red)

no rules are available for this phrase and it was chosen by the phrase-based decoder (purple)

this phrase was chosen by the phrase-based decoder (blue)

156 Exploring an MT Model DeNeefe et al., ACL Demo Session, 2005

Translation Visualization: Spanish

7:30 PM Christopher: es realmente grande para poder hablar a mis colegas en américa latina con este sistema

7:34 PM Adrianna: Si , yo estoy de acuerdo , esto hará que la compañía sea mucho mas eficiente .

7:36 PM Christopher: me siento , pero julio el ingresos informe va a ser de tres semanas tarde que llegan a su oficina

7:37 PM Adrianna: Eso no es un problema , pero usted pudo hacer las correcciones que yo le pedi que hiciera en la nueva sección del informe

7:38 PM Christopher: no yet, he sido llegar los preparativos para la banff conferencia

7:40 PM Adrianna: Usted debiera haber estado revisando el informe más que preocuparse por la próxima reunión en Banff

7:41 PM Christopher: que planificación informe sólo vacío de acción ; hay que poner en marcha para producir un resultado

7:43 PM Adrianna: De acuerdo , entonces yo voy a trabajar en la estrategia de implementación y le enviaré una copia más tarde

Adrianna:  Send

157
Uncertainty in statistical NLP
Collins et al., EuroVis, 2007

# Want more?

[www.infosthetics.com](http://www.infosthetics.com)

[www.visualcomplexity.com](http://www.visualcomplexity.com)

## *Visualization Websites*



# Many Eyes

161

The screenshot shows the Many Eyes website interface. On the left, there are navigation menus for 'explore' (visualizations, data sets, comments, topic hubs), 'participate' (register, create visualization, upload data set, create topic hub), and 'learn more' (quick start, visualization types, about Many Eyes, blog). The main content area is titled 'Try Our Featured Visualizations' and includes four featured items: 'US government expenses 1962-2004', 'University of Michigan faculty salaries', 'Overweight Adults per Country', and 'Privacy Policies'. Below this is a 'Featured Topic Hubs' section with three hubs: 'Food Safety' (illness statistics, food recalls and alerts, etc.), 'Transportation' (Planes, trains, and automobiles), and 'OECD Factbook 2007' (Official statistics). The bottom of the page features the 'many eyes beta' logo, the tagline 'for shared visualization and discovery', the IBM logo, and the URL 'http://www.many-eyes.com'.

<http://www.many-eyes.com>

# Many Eyes

162

- 16 visualization types
- Upload plain text, comma or tab delimited
- Discuss and share visualizations with colleagues
- 2 language-specific visualizations (tag cloud and word tree)
- Comparison views, e.g. change treemap
- Interactive, save any state of view in threaded discussion
- [www.many-eyes.com](http://www.many-eyes.com)

many eyes

Visualizations : Line Graph of F-Score of posts in Robert Scoble's blog (fixed)

Created by: Cornelius Puschmann Created on: Thursday February 01, 3:50 PM

Aggregate items with same label: Average

Data file: F-Score of posts in Robert Scoble's blog (fixed) Data source: Corporate Blogging Corpus

share this watch this add to topic hub rate this

F = 0.5 \*  
 ( (NOUNS + ADJECTIVES +  
 PREPOSITIONS + DETERMINERS)  
 - (PRONOUNS + VERBS + ADVERBS +  
 INTERJECTIONS)  
 + 100)

"If you have a look at those posts, you'll probably notice that they aren't really in any way more *formal* than Scoble's other writing. The difference is that they tend to be more *informational*, i.e. have more and more condensed information crammed into them than most entries."

163 Formality in Blogging CorpBlawg, Cornelius Puschmann, 2008; ManyEyes, Wattenberg et al., 2007

many eyes

Visualizations : Scatterplot of F-score, standard deviation and post frequency in web logs (4)

Created by: Cornelius Puschmann Created on: Thursday February 08, 11:41 AM

X Axis: fscore Y Axis: posts Dot Size: posts

Data file: F-score, standard deviation and post frequency in web logs (4) Data source: Corporate Blogging Corpus

share this watch this add to topic hub rate this

User Profile Link

164 Comparison of Corporate Blogs CorpBlawg, Cornelius Puschmann, 2008; ManyEyes, Wattenberg et al., 2007

# Swivel

165

The screenshot shows the Swivel website interface. At the top, there is a navigation menu with links for Home, Graphs, Data, People, Groups, and Upload. Below this, there are category links: Economics, Entertainment, Health, Politics, Science, Society, Sports, Technology, Miscellaneous, and Official Source. The main content area features a section titled "Upload and explore data." with a sub-section "Spotlight YouTube Generation". This section contains a horizontal bar chart comparing the percentage of viewers ages 18-29 and All video viewers across various video sources. The chart shows that YouTube is the most popular source for both groups, with 49% for young adults and 27% for all viewers. Other sources include MySpace, Don't know/refused, Google Video, Yahoo, Cable or network, News websites, AOL Video, and iTunes. Below the chart, there is a text block explaining the data source (PEW) and providing context about young adults' preferences. To the right of the main content, there are two sidebars: "Swivel Business" with a link to "Swivel Business?" and "Swivelicious Bloggers" with a list of blog links and a "See more blogs >" link. At the bottom right, there is a "12157 Data Sets" section with a link to "Webster County MS Demographics: Population, Crime Rate" and a "See more data >" link. The website URL "www.swivel.com" is visible at the bottom right.

Venue	Viewers ages 18-29 (%)	All video viewers (%)
YouTube	49	27
Other	~15	~15
MySpace	~10	~10
Don't know/refused	~10	~10
Google Video	~5	~5
Yahoo	~5	~5
Cable or network	~5	~5
News websites	~5	~5
AOL Video	~5	~5
iTunes	~5	~5

# Swivel

166

- ❑ Excel-type charts only
- ❑ Discussion forum
- ❑ Comma, tab delimited data upload

# Programming Libraries

167

## prefuse

168

**prefuse**  
INFORMATION VISUALIZATION TOOLKIT

Home | Download | Gallery | Documentation | FAQ

Download  
prefuse beta  
Release 2007-10-23  
source zip (26) 4.1mb

Gallery

the prefuse visualization toolkit

Prefuse is a set of software tools for creating rich interactive data visualizations. The original **prefuse** toolkit provides a visualization framework for the Java programming language. The **prefuse flare** toolkit provides visualization and animation tools for ActionScript and the Adobe Flash Player.

Prefuse supports a rich set of features for data modeling, visualization, and interaction. It provides optimized data structures for tables, graphs, and trees, a host of layout and visual encoding techniques, and support for animation, dynamic queries, integrated search, and database connectivity. Prefuse is written in Java, using the Java 2D graphics library, and is easily integrated into Java Swing applications or web applets. Prefuse is licensed under the terms of a **BSD license**, and can be freely used for both commercial and non-commercial purposes.

The **visualization gallery** and **demonstration video** provide numerous examples of the types of applications that can be built with the prefuse toolkit. To learn more about prefuse, take a look at the **user's manual** or the **frequently asked questions**. For users of the alpha version of the toolkit, there is also a **porting guide** for migrating to the beta version.

Need help? Visit the **Help Forum on Sourceforge.net** (You'll need a Sourceforge login to post). Please be sure to include detailed information (e.g., stack traces, source code, etc) if you need debugging help.

If you are interested in tools for ActionScript and Flash, see the **prefuse flare** project instead.

announcements

**2008.04.02:** Our friends at the IBM Visual Communication Lab are using **prefuse flare** to create visualizations for the **Many-Eyes** visualization service. Check out their new **Comparison Tag Clouds**, made with Flare!

**2007.10.22:** We're happy to announce the first alpha release of **prefuse flare**, a new prefuse-based visualization library written in ActionScript 3! Flare brings the visualization capabilities of prefuse to the web and runs in the Adobe Flash Player.

**2007.06.11:** The Toronto Star, Canada's most highly circulated daily, just ran a story on the **prefuse-based DocuBurst visualization!** Congrats to Chris, the author of DocuBurst! Check out the **prefuse gallery** for DocuBurst, and other great visualization projects.

**2007.02.11:** A number of new projects have been added to the **prefuse gallery**. Check them out!

**2006.05.18:** The **prefuse.org** website has moved to a new server, with better performance and new features. For example, you can now add comments to pages of the user manual. Apologies to any visitors who have encountered 404 errors by using outdated **prefuse.sourceforge.net** URLs.

**2006.04.16:** Prefuse has now surpassed 10,000 downloads! Thanks to everyone who has contributed to the toolkit along the way.

**2006.03.03:** The prefuse-based Vizster visualization appeared on the CBS crime drama **Numb3rs!** Watch the **video clip (WHV, 4.7M)**.

Feeds (Atom | RSS)

releases

**2007.10.21:** prefuse-beta 2007.10.21 released. See the **release notes** for more.

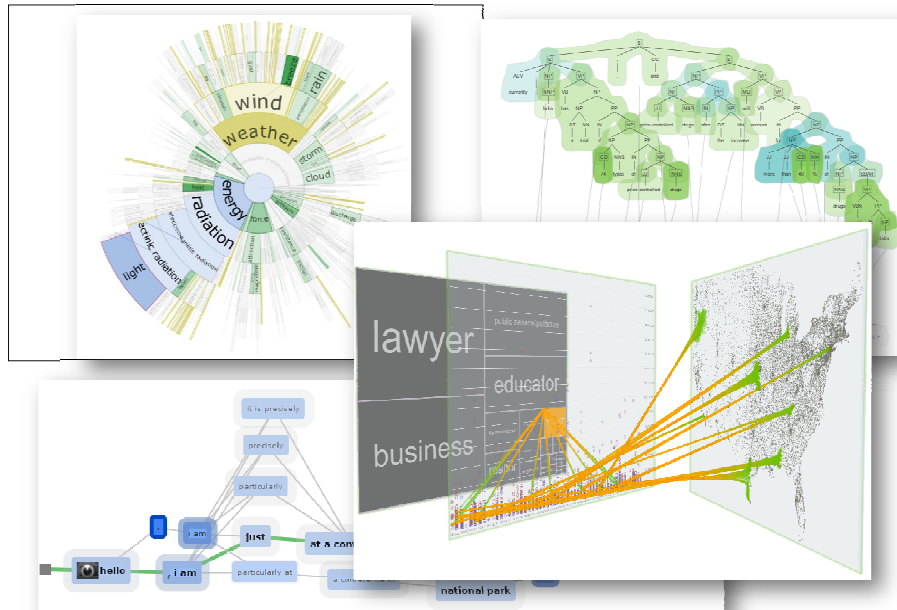
**2008.04.02:** flare-alpha 2008.04.02 released. See the **release notes** for more.

Prefuse.org  
Jeffrey Heer

# prefuse

169

- Open source Java programming library
- BSD license
- Software architecture follows sense-making cycle
  - ▣ Standard data formats supported (I/O)
  - ▣ Interaction out-of-the-box
  - ▣ Supplied collection of layouts and renderers
- Active user support forums
- Relatively fast prototyping
- Easily link with java NLP code libraries
- Our tool of choice!

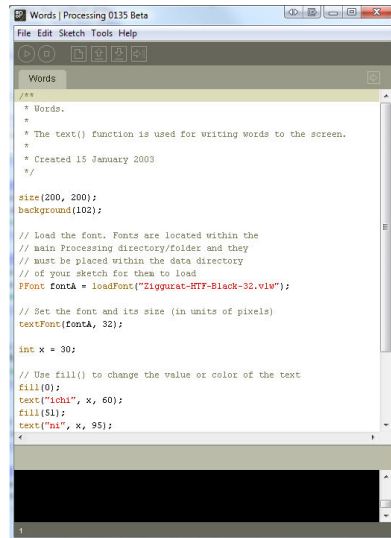


170

prefuse examples by Collins et al.

# Processing

171



```
Words | Processing 0135 Beta
File Edit Sketch Tools Help

Words
/*
 * Words.
 *
 * The text() function is used for writing words to the screen.
 *
 * Created 15 January 2003
 */

size(200, 200);
background(102);

// Load the font. Fonts are located within the
// main Processing directory/folder and they
// must be placed within the data directory
// of your sketch for them to load
PFont fontA = loadFont("Tiggurat-HP-Black-32.vlw");

// Set the font and its size (in units of pixels)
textFont(fontA, 32);

int x = 30;

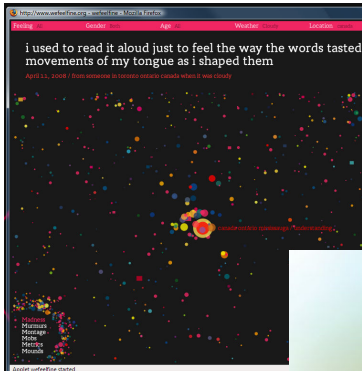
// Use fill() to change the value or color of the text
fill(0);
text("ich!", x, 60);
fill(51);
text("ni", x, 95);
```

Processing.org  
Ben Fry and Casey Reas

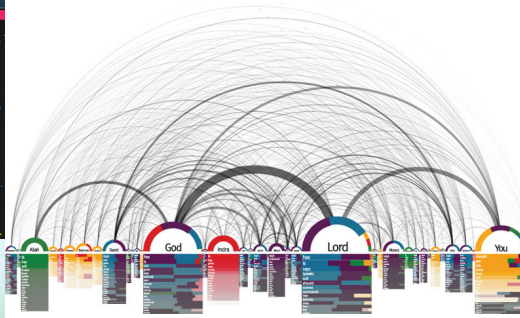
# Processing

172

- ❑ Open source programming language and IDE
- ❑ Simplified graphics and interaction (2D & 3D)
- ❑ Based on Java, can import Java packages
- ❑ Easy to learn
- ❑ Many help resources (online and print)
- ❑ Very wide programmer base
  - mostly designers, artists, students



Jonathan Harris,  
[www.wefeelfine.org](http://www.wefeelfine.org)

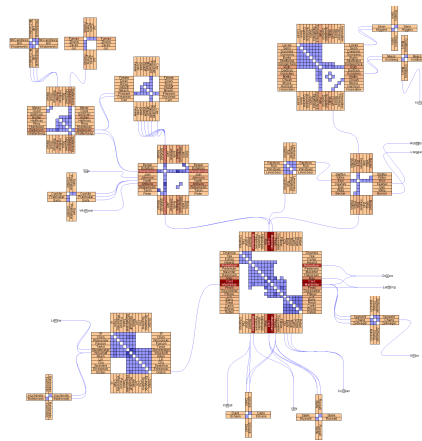


Steinweber & Koller, [similardiversity.net](http://similardiversity.net), 2008



Neumann et al., KeyStrokes, EuroVis 2007

## InfoVis Toolkit

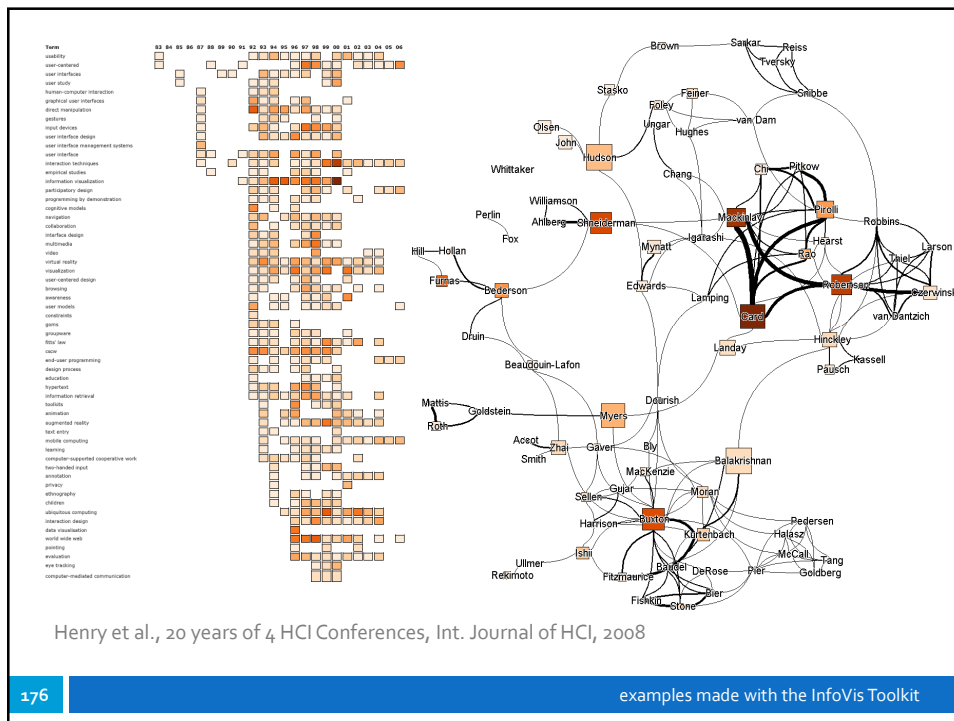


[ivtk.sourceforge.net](http://ivtk.sourceforge.net)  
Jean-Daniel Fekete

# InfoVis Toolkit

175

- Graphics toolkit for Java
- Software architecture follows sense-making cycle
  - Standard data formats supported (I/O)
  - Supplied collection of layouts and renderers
- Matrix and parallel coordinates visualizations maybe especially useful for NLP
- Fast, small memory footprint





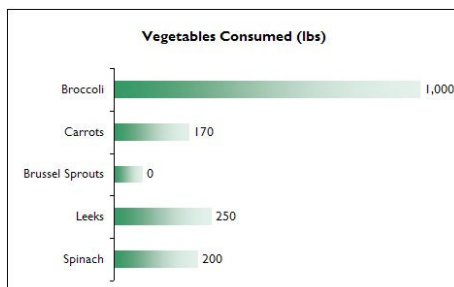
## *Visualization Software*

177

## Microsoft Excel

178

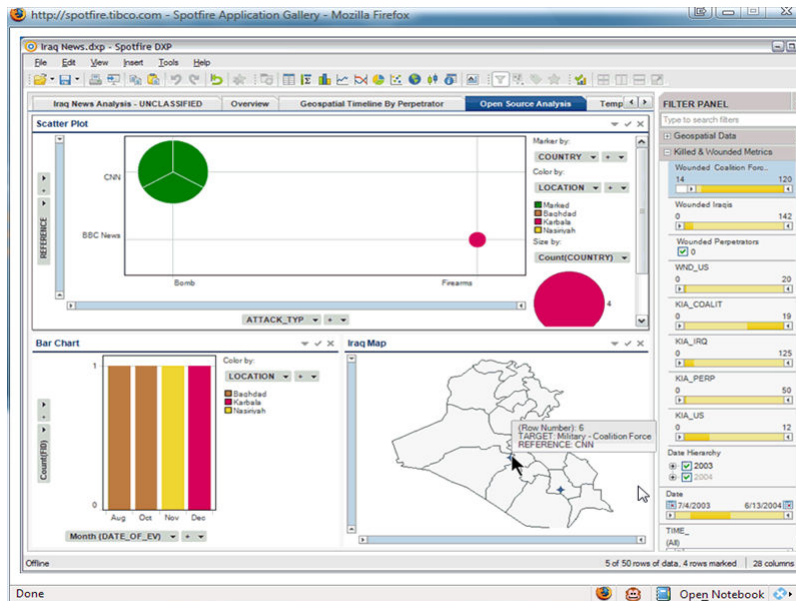
- Can do more than you think!
- ... But be careful, new features can be misleading!
- [www.juiceanalytics.com](http://www.juiceanalytics.com) has great tips



# SpotFire

179

- Coordinated views
- Multi-dimensional data
- Customized for business intelligence, but applicable to quantitative research
- [spotfire.tibco.com](http://spotfire.tibco.com)



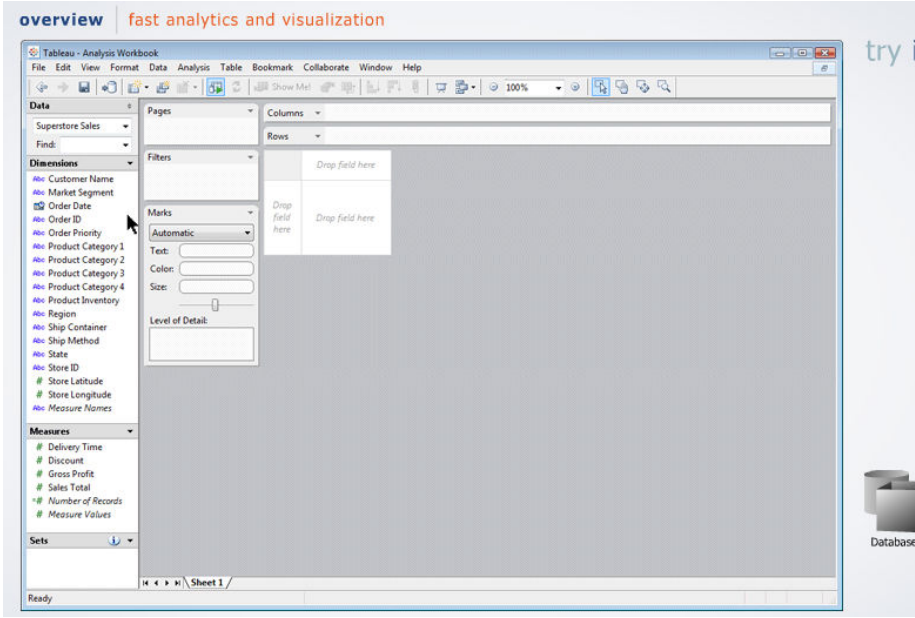
180

[spotfire.tibco.com](http://spotfire.tibco.com)

# Tableau

181

- Easily compose visualizations with “VizQL” language
- Drag and drop data columns into a library of visualizations
- Create “dashboards” of always-up-to-date data graphics
- [tableausoftware.com](http://tableausoftware.com)



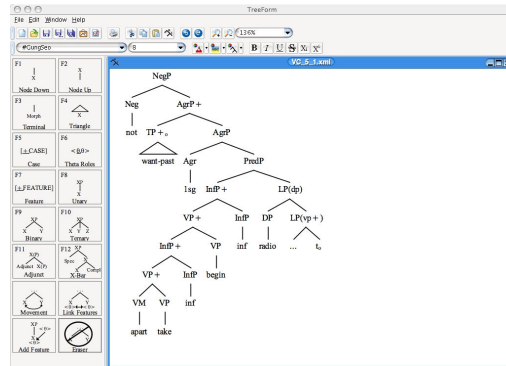
The screenshot displays the Tableau software interface. At the top, there is a navigation bar with 'overview' and 'fast analytics and visualization'. Below this is the main window titled 'Tableau - Analysis Workbook'. The interface is divided into several panes: 'Data' on the left, 'Columns' and 'Rows' at the top, and a central visualization area. The 'Data' pane is expanded to show 'Dimensions' and 'Measures'. The 'Dimensions' list includes fields like Customer Name, Order Date, Order ID, Order Priority, Product Category 1-4, Product Inventory, Region, Ship Container, Ship Method, State, Store ID, Store Latitude, Store Longitude, and Measure Names. The 'Measures' list includes Delivery Time, Discount, Gross Profit, Sales Total, Number of Records, and Measure Values. The 'Columns' and 'Rows' shelves are currently empty, with a 'Drop field here' prompt. The 'Marks' shelf is set to 'Automatic'. The status bar at the bottom indicates 'Ready' and 'Sheet 1 /'. The Tableau logo and 'Database' icon are visible in the bottom right corner.

182

[tableausoftware.com](http://tableausoftware.com)

# TreeForm

183



- Customized software for syntax tree drawing
- Open source initiative lead by Donald Derrick and Daniel Archambault
- Lots of alternative packages, most of them not very good

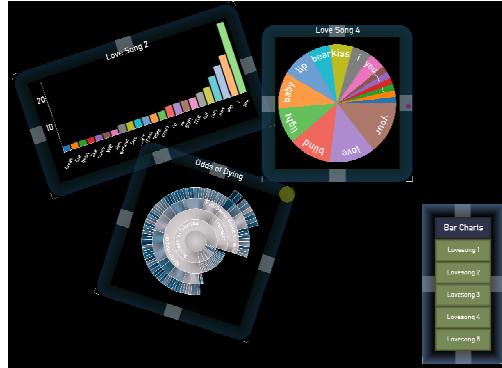
<http://www.ece.ubc.ca/~donaldd/treeform.htm>

## *Emerging Research*

184

# Collaborative Visualization

185



Isenberg and Carpendale, InfoVis 2007

# Toolkits for Visualization on the Web

186

**flare**  
VISUALIZATION ON THE WEB

Home | Tutorial | API Documentation

Download

flare alpha  
release 2008.04.02  
source zip (.zip)  
1.1mb

Demos

flare demos

Tools

Flex SDK  
free actionscript  
compiler  
from adobe systems,  
inc.

Flex Builder  
AS3 devel.  
environment  
from adobe systems,  
inc.

the flare visualization toolkit

ActionScript 3 libraries for interactive visualizations on the web.

Flare is a collection of ActionScript 3 classes for building a wide variety of interactive visualizations. For example, flare can be used to build basic charts, complex animations, network diagrams, treemaps, and more. Flare is written in the ActionScript 3 programming language and can be used to build visualizations that run on the web in the Adobe Flash Player. Flare applications can be built using the free Adobe Flex SDK or Adobe's Flex Builder IDE. Flare is based on **prefuse**, a full-featured visualization toolkit written in Java. Flare is open source software licensed under the terms of the **BSD license**, and can be freely used for both commercial and non-commercial purposes.

Take a look at our initial **flare demo reel** to see some of the visualizations that flare makes it easy to build.

To get up and running with flare, take a look at the **Flare Tutorial** and the **API documentation**.

Need help? Visit the **Flare Help Forum on SourceForge.net** (You'll need a SourceForge login to post). Please be sure to include detailed information (e.g., stack traces, source code, etc) if you need debugging help.

Flare is just getting up and running, so please excuse any rough edges you may encounter, and look for more changes in the near future!

announcements

**2008.04.02:** Our friends at the **IBM Visual Communication Lab** are using Flare to create visualizations for the **Many-Eyes visualization service**. Check out their new **Comparison Tag Clouds**, made with Flare!

**2008.02.19:** A new flare release is now available.

**2007.10.22:** The first alpha release for prefuse flare and the launch of flare.prefuse.org!

releases

**2008.04.02:** flare-alpha 2008.04.02 released.

Flare Visualization Toolkit, Heer 2008

*CL Expertise for InfoVis*

## Improving Document Visualization

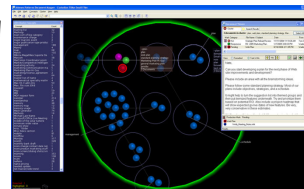
189

- Incorporating WSD, detection of multi-word entities, idioms
- Enabling cross-language comparisons
- Document “difference” visualizations on a semantic level
- Deriving document structure to aid document navigation
- Abstracting document visualization to a level useful and usable for information retrieval (next generation search engine interface)

## e-Discovery

190

- A specialized form of document visualization for lawyers:
  - ▣ Thousands of documents classified individually
  - ▣ Clustering speeds things up drastically
- More accurate keyword detection
- Auto-classification with measures of confidence
- ... Very profitable sector already!

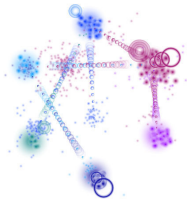


Attenex.com, 2008

## Navigating Email and IM Chat

191

- Existing visualizations use only surface characteristics (letter/word counts, punctuation, meta-data)
- Imagine navigating your email/chat history thematically



BubbaTalk (Tat and Carpendale, 2002)

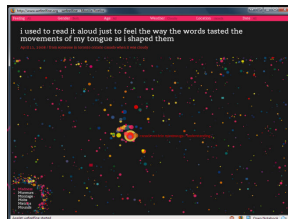


Thread Arcs (Kerr, 2003)

## Managing Streaming Data

192

- RSS feeds from news and blogs
- Facebook/Twitter updates
- Academic journals/library update services
- Social vis community is very active here, appropriating whatever CL methods they can figure out!





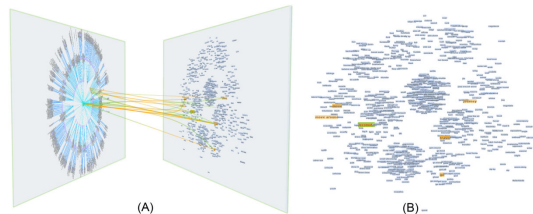
## *InfoVis to Further CL Research*

193

## Structural Comparisons

194

- Visualization to show similarities and differences in data structures:
  - ▣ Comparing parse trees and parse representations
  - ▣ Comparing ontologies, other knowledge sources
  - ▣ Language change over time
  - ▣ Lexical semantic distance measures
  - ▣ Others?



# Exploratory Data Analysis

195

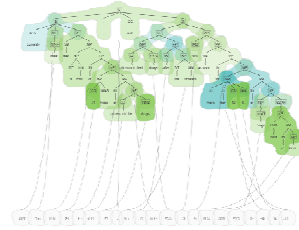
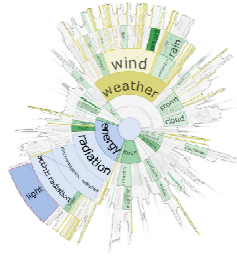
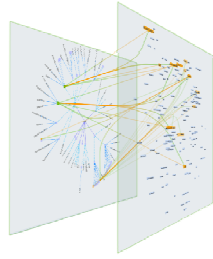
- Visualizing corpora
  - Quality control
  - Deep investigation of inter-annotator agreements
  - Discover areas of imbalanced data coverage
- Interactive exploration of parameter spaces
  - “What changes when I adjust this parameter?”



# Understanding NLP Processes

196

- “Live” visualization of automata
  - Dialogue system construction
  - Visualizing non-determinism
- Visualizing uncertainty in parametric models
- Visualization of chart pruning and beam search
- Hypothesis tracking
  - Machine translation
  - Speech recognition
- Others?



*<http://www.infovis-wiki.net> → Research & Education → Linguistic Visualization  
or Search "linguistic visualization wiki"*

CHRISTOPHER COLLINS  
CCOLLINS@CS.UTORONTO.CA

GERALD PENN  
GPENN@CS.UTORONTO.CA

SHEELAGH CARPENDALE  
SHEELAGH@UCALGARY.CA



# Annotated Bibliography

## References

- [1] C. Ahlberg and B. Shneiderman, “The alphslider: A compact and rapid selector,” in *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, 1994, pp. 365–371.
- [2] J. Analytics, “Juice analytics blog,” Website, 2008. [Online]. Available: <http://www.juiceanalytics.com/writing>
- [3] F. J. Anscombe, “Graphs in statistical analysis,” *American Statistician*, vol. 27, pp. 17–21, Feb. 1973.
- [4] L. Arent, A. Logan, and G. Havin, “Using color in information display graphics,” Website, May 2008. [Online]. Available: <http://colorusage.arc.nasa.gov>  
  
Website providing practical advice on creating readable, usable colour schemes for information graphics. No longer maintained.
- [5] J. Bertin, *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press, 1983.  
  
Currently out of print, this text contains the best discussion of ‘visual variables’ for clear communication of printed information graphics.
- [6] C. Brewer and M. Harrower, “Colorbrewer,” Website. [Online]. Available: <http://www.colorbrewer.org>  
  
Interactive website for designing and testing good color schemes for maps and graphics.
- [7] S. K. Card, J. D. Mackinlay, and B. Shneiderman, *Readings in Information Visualization: Using Vision to Think*. San Francisco, USA: Morgan Kaufmann, 1999.  
  
A much-read compilation of early influential papers in the area of interactive information visualization. Introductory essay discusses a well-accepted model for the visual analysis process.
- [8] M. Carpendale, “Considering visual variables as a basis for information visualisation,” Department of Computer Science, University of Calgary, Calgary, Canada, Tech. Rep. 2001-693-16, 2003. [Online]. Available: [http://pharos.cpsc.ucalgary.ca/Dienst/Repository/2.0/Body/ncstrl.ucalgary\\_cs/2001-693-16/pdf](http://pharos.cpsc.ucalgary.ca/Dienst/Repository/2.0/Body/ncstrl.ucalgary_cs/2001-693-16/pdf)

A helpful review of Bertin's set of 'visual variables', especially as Bertin's text is out-of-print. This report extends the set of variables to propose additional encoding methods introduced when visualization moves from paper to the screen.

- [9] J. Clark, "Neoformix: Discovering and illustrating patterns in data," Website, May 2008. [Online]. Available: <http://www.neoformix.com>
- [10] W. S. Cleveland and R. McGill, "Graphical perception: Theory, experimentation, and application to the development of graphical methods," *Journal of the American Statistical Association*, vol. 79, no. 387, pp. 531–554, Sept. 1984.

A good discussion of visual variables, including ideas about which variables may be more appropriate for quantitative, ordinal, and nominal data.

- [11] C. Collins, "Docuburst: Radial space-filling visualization of document content," Knowledge Media Design Institute, University of Toronto, Tech. Rep. KMDI-TR-2007-1, 2007.
- [12] C. Collins and S. Carpendale, "VisLink: Revealing relationships amongst visualizations," *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization)*, vol. 13, no. 6, Nov./Dec. 2007.

Introduces a visual framework for linking multiple 2D visualizations in a constrained 3D space, providing additional analytical power through inter-visualization queries and revealing patterns of similarity and difference between visualizations.

- [13] C. Collins, S. Carpendale, and G. Penn, "Visualization of uncertainty in lattices to support decision-making," in *Proc. of Eurographics/IEEE VGTC Symposium on Visualization (Euro Vis)*. Eurographics, May 2007.
- [14] P. DeCamp, A. Frid-Jimenez, J. Guinness, and D. Roy, "Gist icons: Seeing meaning in large bodies of literature," in *Proc. of IEEE Symp. on Information Visualization, Poster Session*, Oct. 2005.
- [15] S. DeNeefe, K. Knight, and H. H. Chan, "Interactively exploring a machine translation model," in *Proc. Annual Meeting of the Assoc. for Computational Linguistics, Poster Session*, 2005.

Tool for exploring the space of options available to a statistical MT decoder. Provides interactive simulation of the algorithm, helping researchers better understand the model and its operations.

- [16] D. Derrick and D. Archambault, "TreeForm," Website and software, 2008. [Online]. Available: <http://www.ece.ubc.ca/~donald/treeform.htm>
- [17] E. Dickinson, *Open Me Carefully: Emily Dickinson's Intimate Letters to Susan Huntington Dickinson*, M. N. Smith and E. L. Hart, Eds. Paris Press, 1998.

Data source for Dickinson examples.

- [18] D. Dimov and B. Mulloy, "Swivel preview," Website, May 2008. [Online]. Available: <http://www.swivel.com>

- [19] A. Don, E. Zheleva, M. Gregory, S. Tarkan, L. Auvil, T. Clement, B. Shneiderman, and C. Plaisant, “Discovering interesting usage patterns in text collections: Integrating text mining with visualization,” in *Proc. of the Conf. on Information and Knowledge Management*, 2007.
- [20] B. Dougherty and A. Wade, “Vischeck,” Website, May 2007. [Online]. Available: <http://www.vischeck.com>
- Website simulates colour blindness with user-supplied images.
- [21] J.-D. Fekete, “The infovis toolkit,” Website and software, Nov. 2005. [Online]. Available: <http://ivtk.sourceforge.net>
- [22] B. Fry and C. Reas, “Processing,” Website and software, May 2008. [Online]. Available: <http://www.processing.org>
- [23] M. L. Gregory, N. Chinchor, P. Whitney, R. Carter, E. Hetzler, and A. Turner, “User-directed sentiment analysis: Visualizing the affective content of documents,” in *Proc. of the Workshop on Sentiment and Subjectivity in Text*. ACL, 2006, pp. 23–30.
- [24] J. Harris and S. Kamvar, “We feel fine,” 2006. [Online]. Available: <http://www.wefeelfine.org/>
- [25] S. Havre, E. Hetzler, P. Whitney, and L. Nowell, “ThemeRiver: visualizing thematic changes in large document collections,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, Jan. 2002.
- [26] C. G. Healey, “Perception in visualization,” Website, 2007. [Online]. Available: <http://www.csc.ncsu.edu/faculty/healey/PP>
- [27] C. G. Healey, K. S. Booth, and J. T. Enns, “High-speed visual estimation using preattentive processing,” *ACM Transactions on Computer-Human Interaction*, vol. 3, no. 2, pp. 107–135, 1996.
- A good discussion of the idea of preattentive processing. This paper presents one side of the argument, in favour of the theory. The subject is controversial, but no matter what the true nature of how various types of visual information are processed, the empirical evidence suggests some visual variables are faster to read than others.
- [28] M. Hearst, “Information visualization: Principles, promise, and pragmatics,” Tutorial Notes from CHI 2003; Accessed online, 2003. [Online]. Available: <http://bailando.sims.berkeley.edu/talks/chi03-tutorial.ppt>
- [29] M. A. Hearst, “Tilebars: visualization of term distribution information in full text information access,” in *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM Press, 1995, pp. 59–66.
- [30] J. Heer, “Exploring enron,” Website, June 2005. [Online]. Available: <http://jheer.org/enron>
- [31] J. Heer, S. K. Card, and J. A. Landay, “prefuse: a toolkit for interactive information visualization,” in *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM Press, Apr. 2005.
- [32] J. Heer and danah boyd, “Vizster: Visualizing online social networks,” in *Proc. of the IEEE Symp. on Information Visualization*, 2005.

- [33] J. Heer and G. Robertson, “Animated transitions in statistical data graphics,” *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization)*, vol. 13, no. 6, pp. 1240–1247, Nov./Dec. 2007.
- [34] P. Isenberg and S. Carpendale, “Interactive tree comparison for co-located collaborative information visualization,” *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization)*, vol. 13, no. 6, pp. 1232–1238, Nov./Dec. 2007.
- [35] J. Kamps, M. Marx, R. J. Mokken, and M. de Rijke, “Using wordNet to measure semantic orientation of adjectives,” in *Proc. of the 4th Annual Conference on Language Resources and Evaluation (LREC)*, 2004, pp. 1115–1118.
- [36] Kartoo. (2005) Kartoo. [Online]. Available: [www.kartoo.com](http://www.kartoo.com)
- [37] S. Kempken, T. Pilz, and W. Luther, “Visualization of rule productivity in deriving non-standard spellings,” in *Proc. of SPIE-IS&T Electronic Imaging (VDA '07)*, vol. 6495, 2007.
- [38] B. Kerr and E. Wilcox, “Designing remain: Reinventing the email client through innovation and integration,” IBM Research, Tech. Rep. RC23127, 2004.
- [39] A. Leuski, C.-Y. Lin, and E. Hovy, “iNeATS: Interactive mult-document summarization,” in *Proc. of the Annual Meeting of the Association for Computational Linguistics*, ser. Interactive Posters and Demos Session, July 2003. [Online]. Available: <http://www.isi.edu/~cyl/papers/iNeATS-ACL2003.pdf>
- [40] A. Leuski, C.-Y. Lin, L. Zhou, U. Germann, F. J. Och, and E. Hovy, “Cross-lingual C\*ST\*RD: English access to Hindi information,” *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 2, no. 3, pp. 245–269, Sept. 2003. [Online]. Available: <http://doi.acm.org/10.1145/979872.979877>
- [41] G. Levin and Z. Lieberman, “In-situ speech visualization in real-time interactive installation and performance,” in *Proc. of the 3rd international symposium on Non-photorealistic animation and rendering*. ACM, 2004, pp. 7–14.
- [42] C. D. Manning, K. Jansz, and N. Indurkha, “Kirrkirr: Software for browsing and visual exploration of a structured walpiri dictionary,” *Literary and Linguistic Computing*, vol. 16, no. 2, pp. 135–151, 2001.
- [43] D. A. Monsef, “Colourlovers,” Website, May 2008. [Online]. Available: <http://www.colourlovers.com>

Inspired more by design than solid research in perception, this community palette-sharing site can offer aesthetic inspiration. But users should exercise caution in that the suggested colour schemes may not be appropriate for data encoding.

- [44] P. Neumann, A. Tat, T. Zuk, and S. Carpendale, “KeyStrokes: Personalizing typed text with visualization,” in *Proc. of Eurographics/IEEE VGTC Symposium on Visualization (EuroVis)*. Eurographics, May 2007.
- [45] D. A. Norman, *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine*. Boston, USA: Addison-Wesley Longman Publishing Co., 1993.

A classic look at the nature and characteristics of human intelligence and how we are aided (and hindered) by the technology in our lives. Norman argues for an external cognition approach where technology complements human abilities.

- [46] W. B. Paley, “TextArc: Showing word frequency and distribution in text,” in *Proc. of the IEEE Symp. on Information Visualization*, ser. Poster. IEEE Computer Society, Oct. 2002. [Online]. Available: <http://www.textarc.org/appearances/InfoVis02/InfoVis02.TextArc.pdf>
- [47] T. Pilz, A. Philipsenburg, and W. Luther, “Visualizing the evaluation of distance measures,” in *Proc. of the ACL SIG in Computational Morphology and Phonology*. ACL, 2007, pp. 84–92.
- [48] C. Plaisant, J. Rose, B. Yu, L. Auvil, M. G. Kirschenbaum, M. N. Smith, T. Clement, and G. Lord, “Exploring erotics in emily dickinson’s correspondence with text mining and visual interfaces,” in *Proc. of the Joint Conference on Digital Libraries*, 2006.
- [49] S. Ploux and H. Ji, “A model for matching semantic maps between languages (French/English, English/French),” *Computational Linguistics*, vol. 29, no. 2, pp. 155–178, June 2003.
- [50] R. Rao and S. K. Card, “The table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information,” in *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, 1994.
- [51] M. Rembold and J. Späth. (2006) Graphical visualization of text similarities in essays in a book. [Online]. Available: <http://www.munterbund.de/visualisierung-textaehnlichkeiten/essay.html>
- [52] R. L. Ribler and M. Abrams, “Using visualization to detect plagiarism in computer science classes,” in *Proc. of the IEEE Symp. on Information Visualization*. IEEE Press, 2000, pp. 173–178.
- [53] G. G. Robertson, J. D. Mackinlay, and S. K. Card, “Cone trees: animated 3d visualizations of hierarchical information,” in *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, 1991, pp. 189–194.
- Early example of the reconfigure operation in interactive visualization.
- [54] Y. Rogers, “New theoretical approaches for HCI,” *Annual Review of Information Science and Technology*, vol. 38, no. 87–143, 2004.
- A review of theoretical approaches in HCI research, including the external cognition approach.
- [55] P. Saraiya, C. North, and K. Duca, “An insight-based methodology for evaluating bioinformatics visualizations,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 4, pp. 443–456, 2005.
- Describes an experimental method for measuring the insight (amount and quality) that can be gained from a given visualization, and results from applying this method in the biological domain.
- [56] B. Shneiderman and C. Plaisant, “Strategies for evaluating information visualization tools: Multi-dimensional in-depth long-term case studies,” in *Proc. of BELIV 2006*, 2006.
- Suggests a methodology for conducting long term, in situ, evaluations of information visualizations through deployment to real end users (data domain experts).



- [57] L. J. Shuman, “Newsglobe,” Website, Feb. 2008. [Online]. Available: <http://next.yahoo.net/archives/93/newsglobe>
- [58] R. Spence, *Information Visualization*. Toronto, Canada: ACM Press, 2001.
- Popular introductory text in information visualization.
- [59] “Spotfire by TIBCO,” Website and software, 2008. [Online]. Available: <http://spotfire.tibco.com>
- [60] P. Steinweber and A. Koller, “Similar diversity,” Website and gallery installation, Apr. 2008. [Online]. Available: <http://www.similardiversity.net>
- [61] M. Stone, “Stonesoup consulting,” Website, May 2008. [Online]. Available: <http://www.stonesc.com>
- Maureen Stone is a widely sought-after expert in the use of colour in data visualization. Her website contains many useful papers, notes, and practical advice.
- [62] M. C. Stone, *A Field Guide to Digital Color*. AK Peters, Ltd., 2003.
- [63] “Tableau software,” Website and software. [Online]. Available: <http://www.tableausoftware.com>
- [64] A. Tat and M. S. T. Carpendale, “Visualising human dialog,” in *Proc. of the Int. Conf. on Information Visualization*, 2002, pp. 16–21.
- [65] N. Thiessen, “Connection maps: A new way to visualize similarity relationships,” Master’s thesis, University of Toronto, 2004.
- [66] ThinkMap, “ThinkMap visual thesaurus,” Apr. 2005. [Online]. Available: <http://www.visualthesaurus.com>
- [67] E. R. Tufte, *The Visual Display of Quantitative Information*, 2nd ed. Cheshire, USA: Graphics Press, 2001.
- This book and the others by Tufte focus mostly on printed diagrams, but offer much practical advice for creating rich, readable, useful information graphics.
- [68] F. B. Viégas, M. Wattenberg, and K. Dave, “Studying cooperation and conflict between authors with history flow visualizations,” in *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM Press, 2004, pp. 575–582.
- [69] F. B. Viégas, M. Wattenberg, J. Kriss, and M. McKeon, “Many eyes: A site for visualization at internet scale,” *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization)*, vol. 13, no. 6, pp. 1121–1128, Nov./Dec. 2007.
- [70] Visual Communication Lab, IBM Research, “Many eyes,” Website, May 2008. [Online]. Available: <http://www.many-eyes.com>
- [71] M. Wattenberg, “Color code,” Website, 2005. [Online]. Available: <http://www.bewitched.com/live/colorcode>
- [72] C. Weaver, D. Fyfe, A. Robinson, D. W. Holdsworth, D. J. Peuquet, and A. M. MacEachren, “Visual exploration and analysis of historic hotel visits,” in *Proc. of the IEEE Symp. on Visual Analytics Science and Technology (VAST)*, 2006.

- [73] M. Weskamp, “Newsmap,” Website, Aug. 2004. [Online]. Available: <http://www.marumushi.com/apps/newsmap/newsmap.cfm>
- [74] J. A. Wise, J. J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow, “Visualizing the non-visual: spatial analysis and interaction with information for text documents,” in *Readings in Information Visualization: Using Vision to Think*, S. K. Card and J. D. Mackinlay, Eds. San Francisco, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 442–450. [Online]. Available: <http://ieeexplore.ieee.org/iel3/4050/11604/00528686.pdf?arnumber=528686>
- [75] J. S. Yi, Y. ah Kang, J. Stasko, and J. Jacko, “Toward a deeper understanding of the role of interaction in information visualization,” *IEEE Transactions on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization)*, vol. 13, no. 6, pp. 1224–1231, Nov./Dec. 2007.

This taxonomy of interaction techniques provides a useful model for understanding the range of available methods. However, several of the suggested techniques blur the line between presentation (views) and interaction. The taxonomy therefore does not fit perfectly with the visual information-seeking model of Card et al.