READINGS IN

INFORMATION VISUALIZATION

USING VISION TO THINK

WRITTEN AND EDITED BY

STUART K. CARD

XEROX PALO ALTO RESEARCH CENTER

JOCK D. MACKINLAY

XEROX PALO ALTO RESEARCH CENTER

BEN SHNEIDERMAN

UNIVERSITY OF MARYLAND

MORGAN KAUFMANN PUBLISHERS, INC.

SAN FRANCISCO, CALIFORNIA

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."[1] 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 34×72, 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.

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

[1]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.

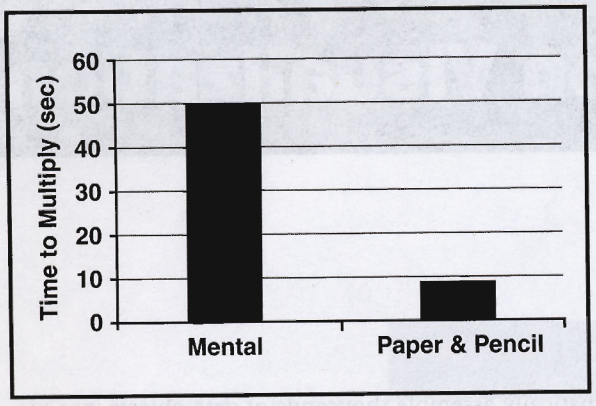


FIGURE **1.1**

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 Tractenberg 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 Tractenberg 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 longhand 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

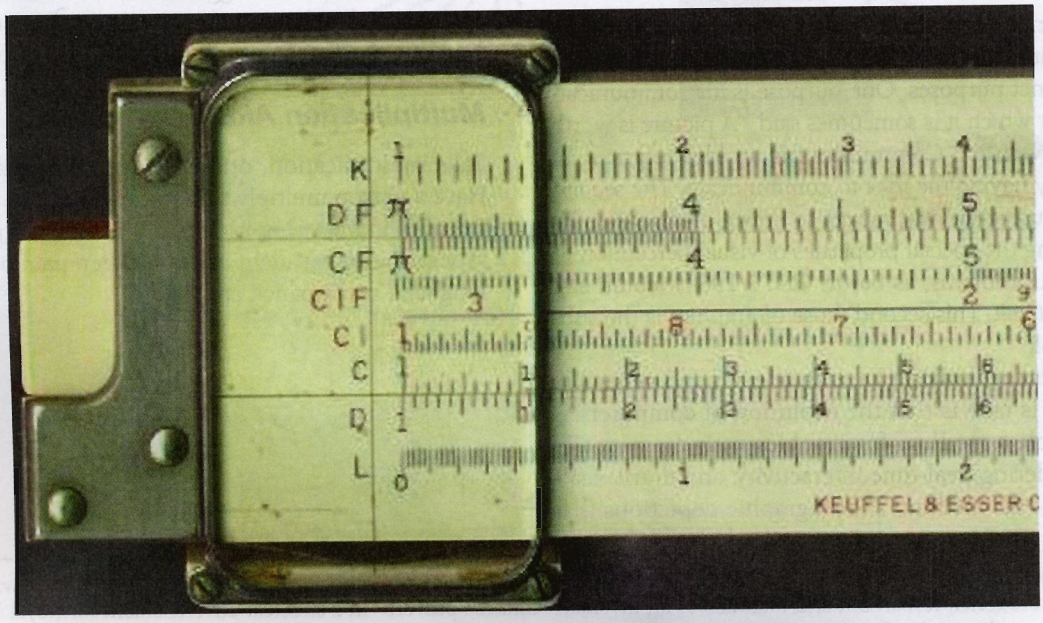


FIGURE **1.2**

Section of a slide rule.

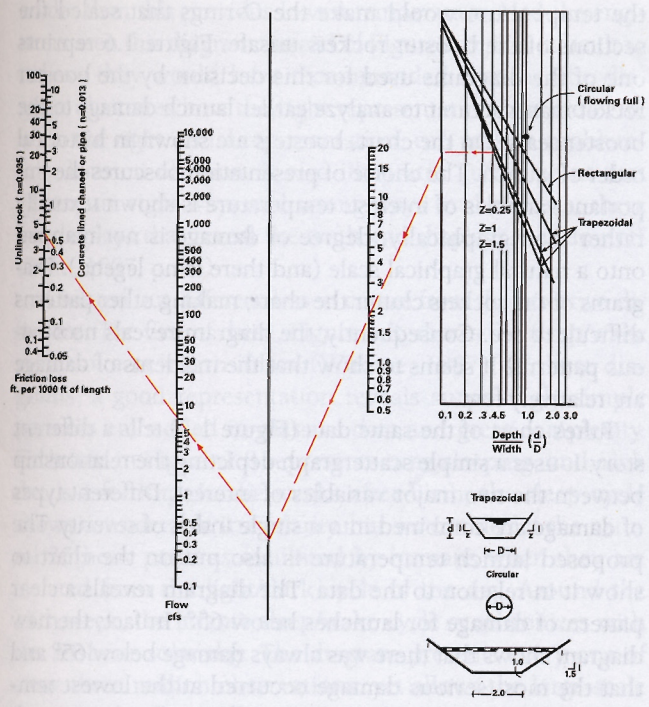


FIGURE **1.3**

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

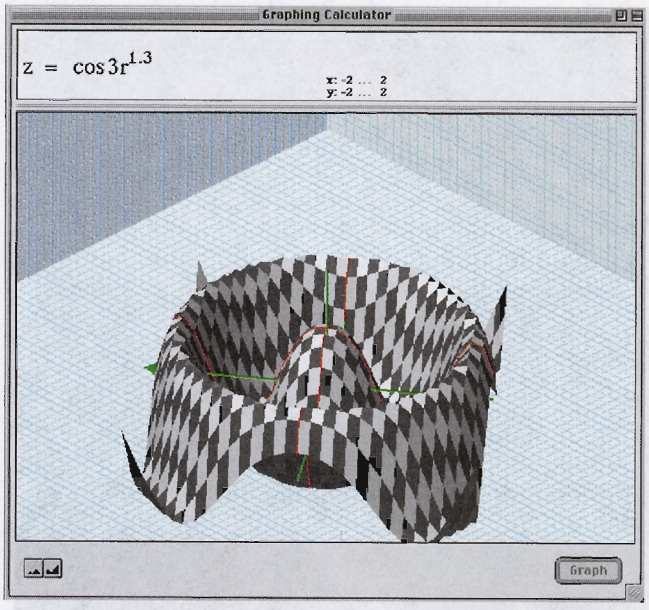


FIGURE **1.4**

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

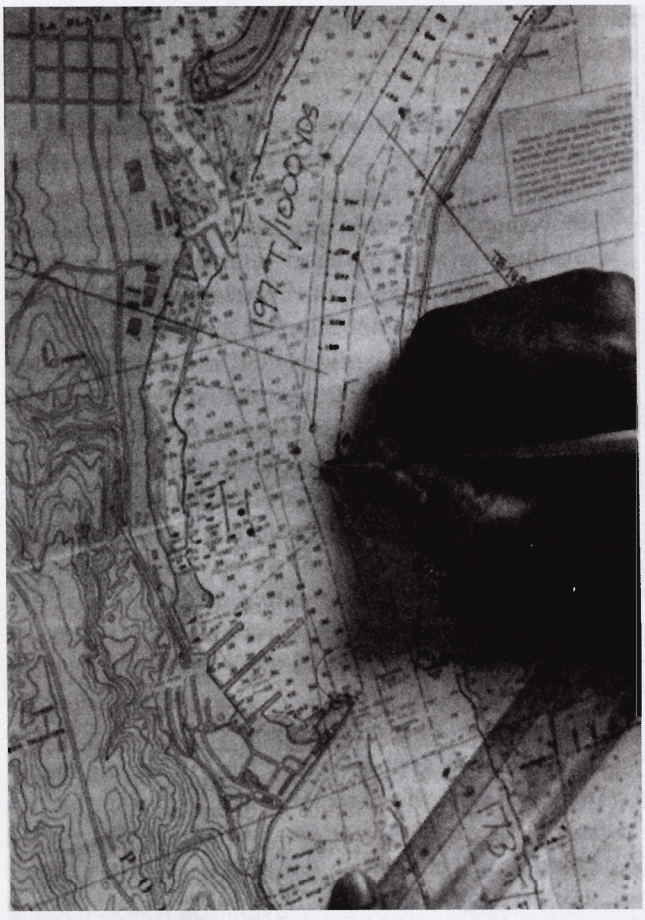


FIGURE **1.5**

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

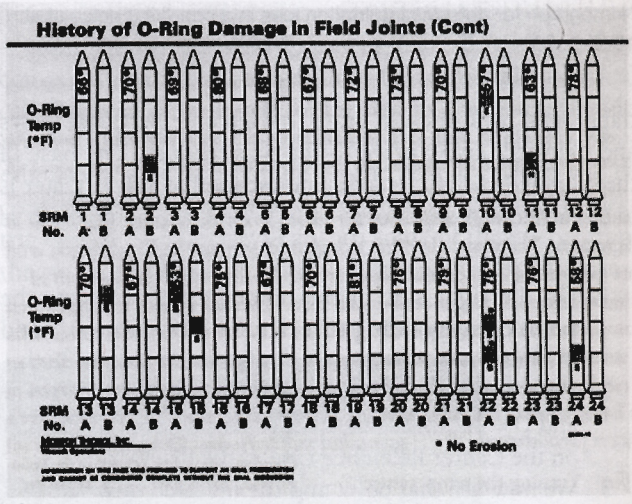


FIGURE **1.6**

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).

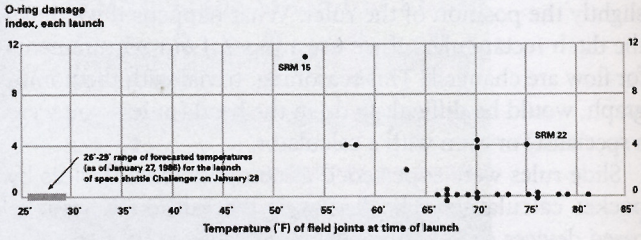


FIGURE **1.7**

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

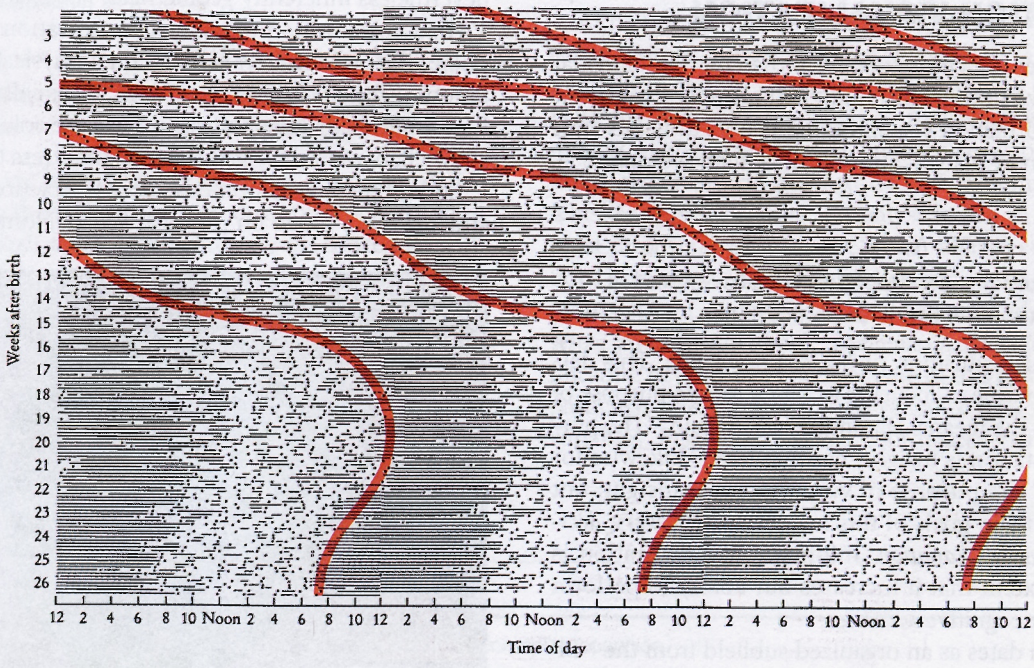


FIGURE **1.8**

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).

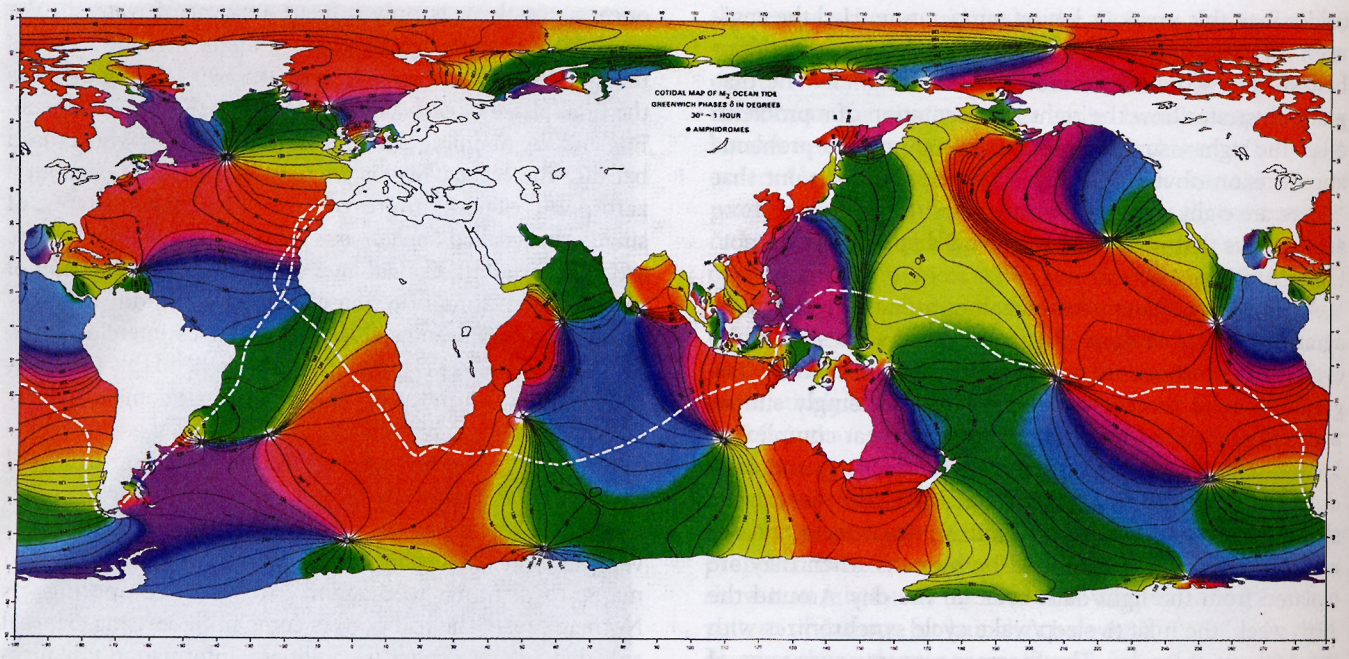


FIGURE **1.9**

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 anphidromic points. The dotted white line shows the route of Magellan's ship (Winfree, 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 visualizations 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-

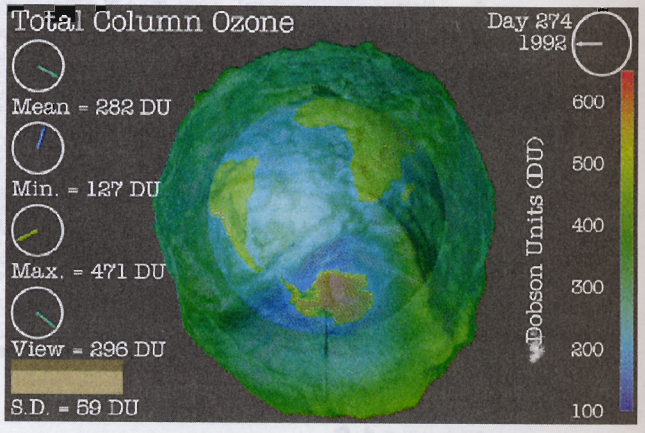


FIGURE **1.10**

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 graphics. *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 *external world* to accomplish cognition.
Information design	Design of *external representations* to amplify cognition.
Data graphics	Use of *abstract, nonrepresentational* visual representations of data to amplify cognition.
Visualization	Use of *computer-based, interactive* visual representations of data to amplify cognition.
Scientific visualization	Use of interactive visual representations of *scientific data,* typically *physically based,* to amplify cognition.
Information visualization	Use of interactive visual representations of *abstract, nonphysically based data* 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.

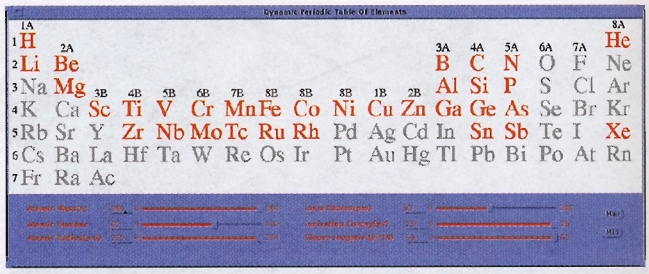


FIGURE **1.12**

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 Mendeleyev, 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 Mendeleyev'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

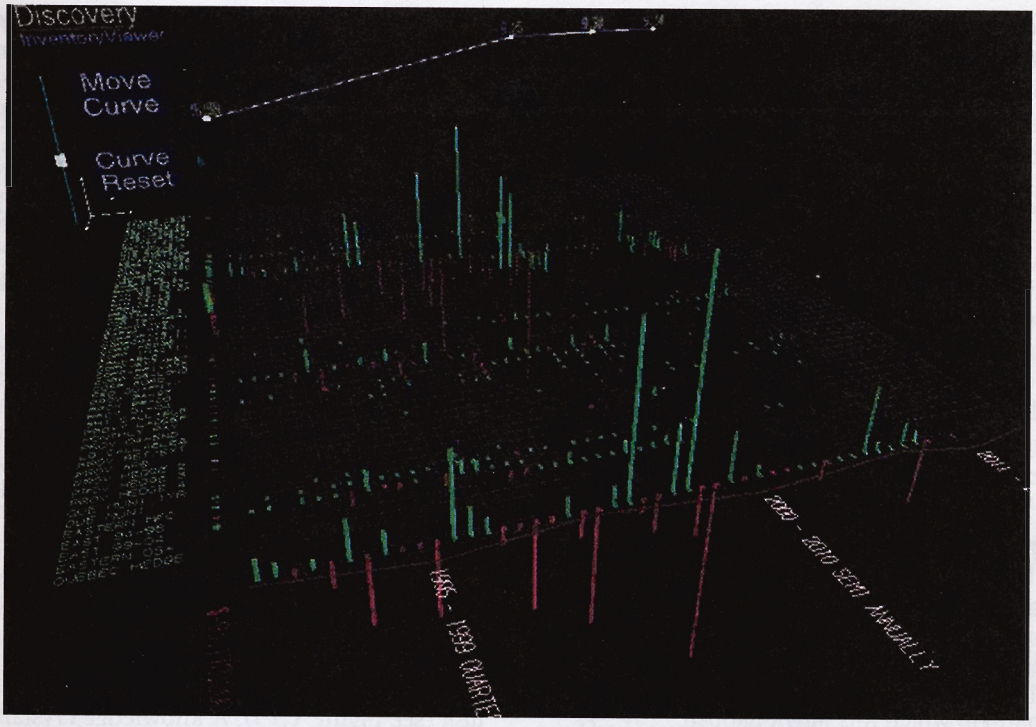


FIGURE **1.13**

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

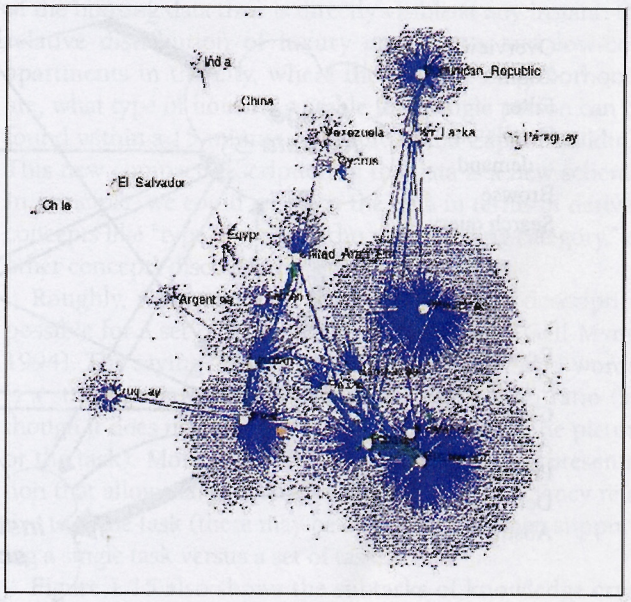


FIGURE **1.14**

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 reordered 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

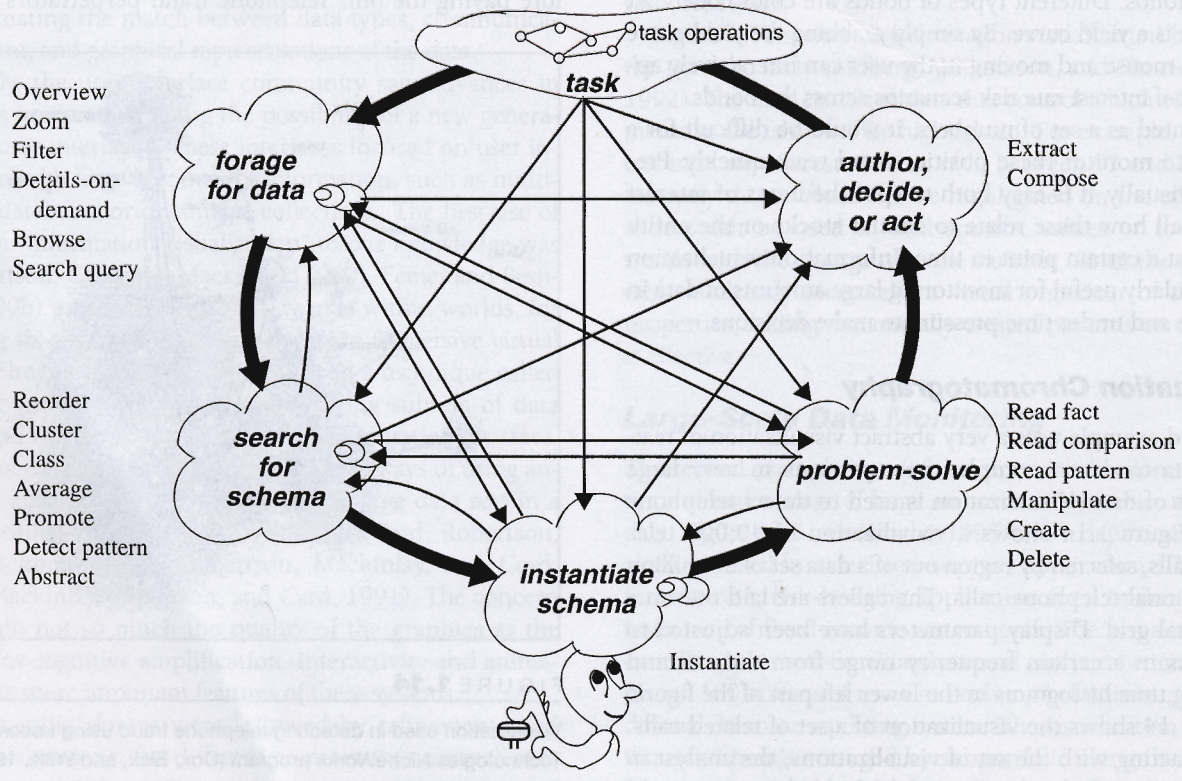


FIGURE **1.15**

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 × 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 responding 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.

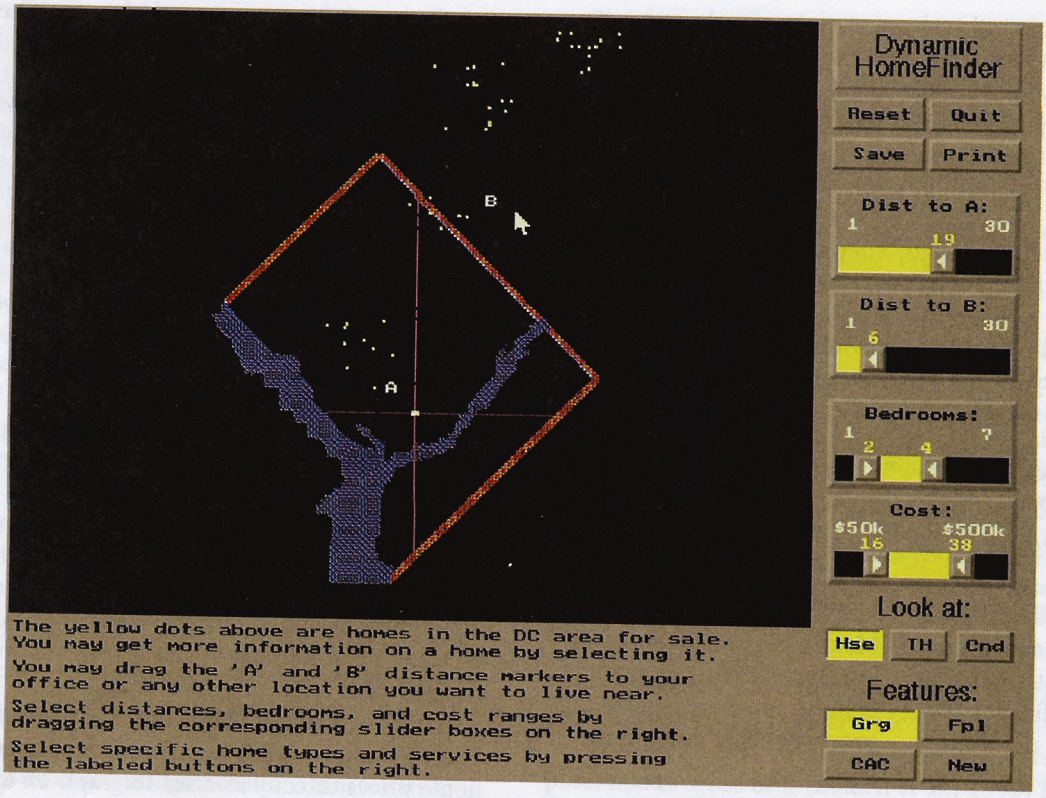


FIGURE **1.16**

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.

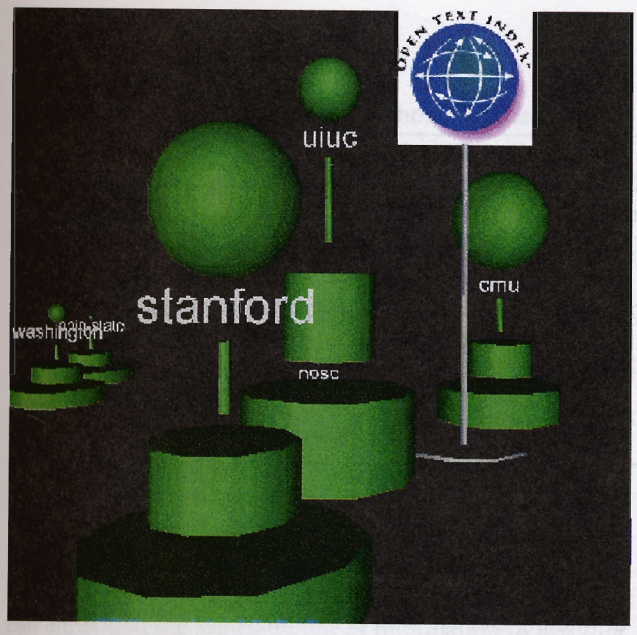
Visualization Levels of Use

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.)

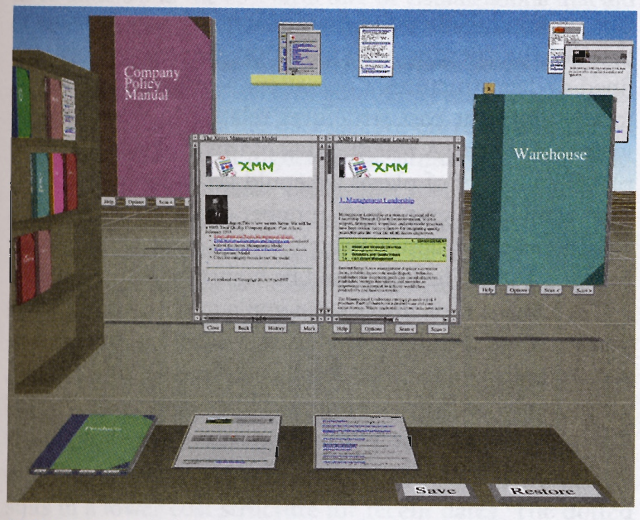
Visualization can be combined with information access techniques to help the user find information. By the *infosphere,* we mean information outside of the user's work environment. This could be information on the World Wide Web, or it could be information in a specific organizational document collection 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.

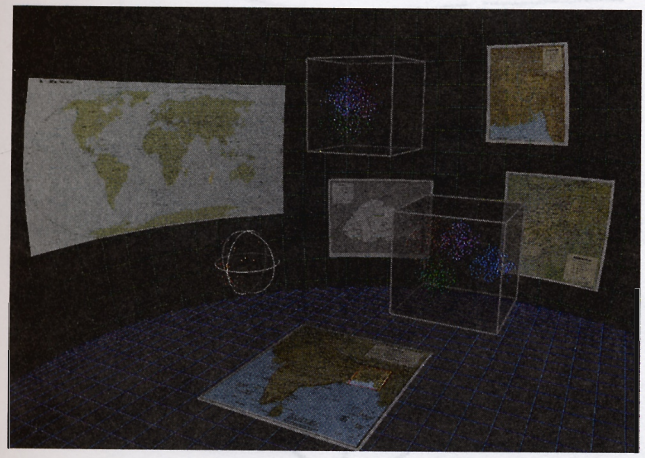
Most visualizations fall at the level of *visual knowledge tools,* as shown in Figure 1.17(d)(e). Either they arrange information to reveal patterns, or they allow the manipulation



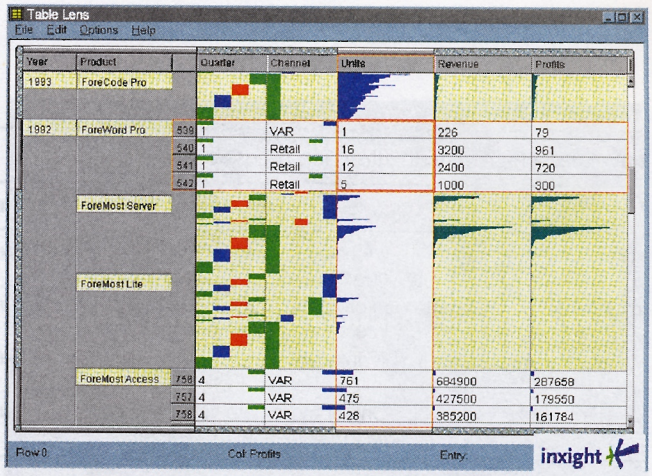
(a) View of sites on the World Wide Web (Bray, 1996 ●, detail from Figure 11).



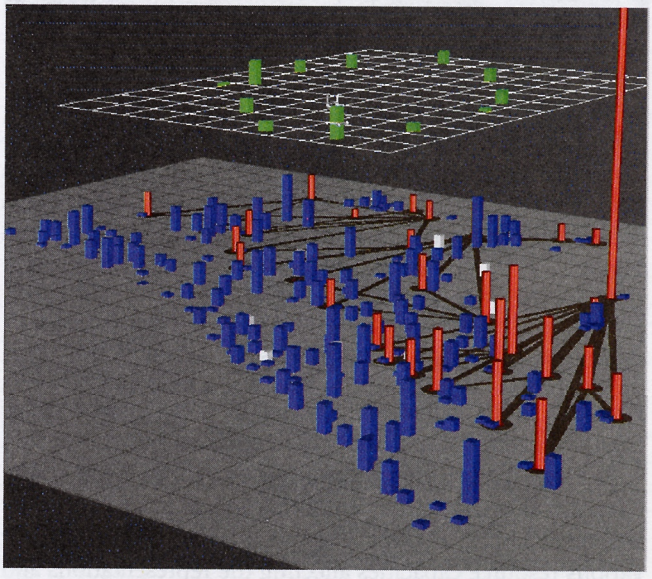
(b) Workspace of Web page. Courtesy of Xerox Corporation. See Card, Robertson, and York (1996 ●).



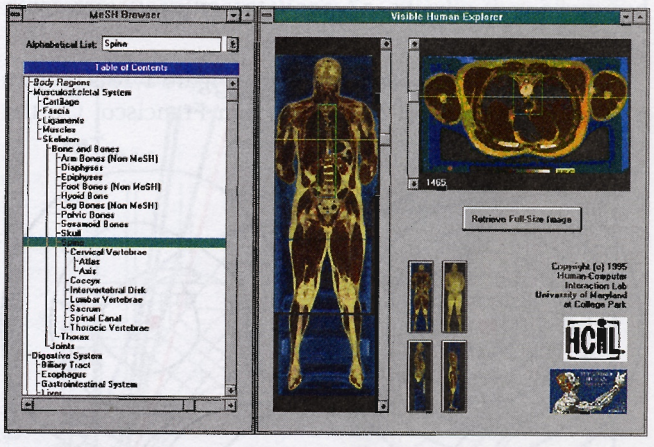
(c) Workspace for document (Risch et al., 1997 ●, detail from Figure 1).



(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.

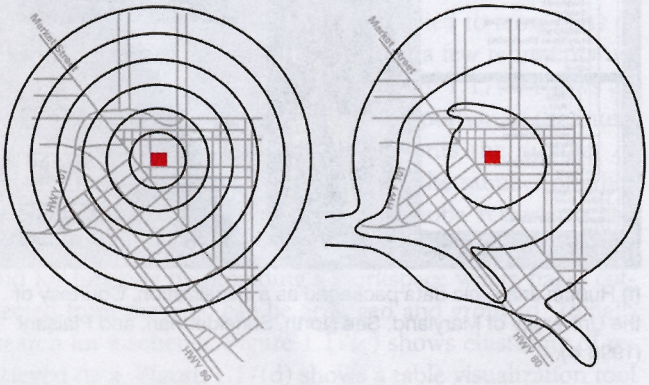
Visualization can also operate at the level of *visually enhanced objects*. These refer to objects, especially virtual physical objects such as the human body or a book, that have been enhanced with visualization techniques to package collections of abstract information. The anatomic browser in Figure 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 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,



Downtown San Francisco, CA
(a)

Downtown San Francisco, CA
(b)

FIGURE **1.18**

Cost structure for driving and walking in San Francisco.

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

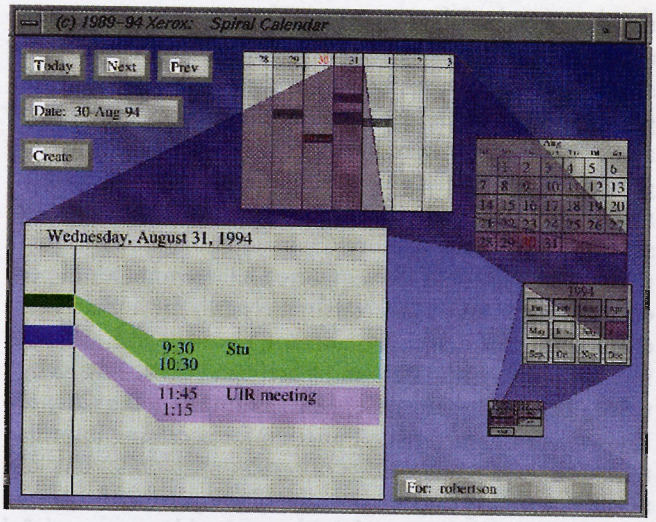


FIGURE **1.21**

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

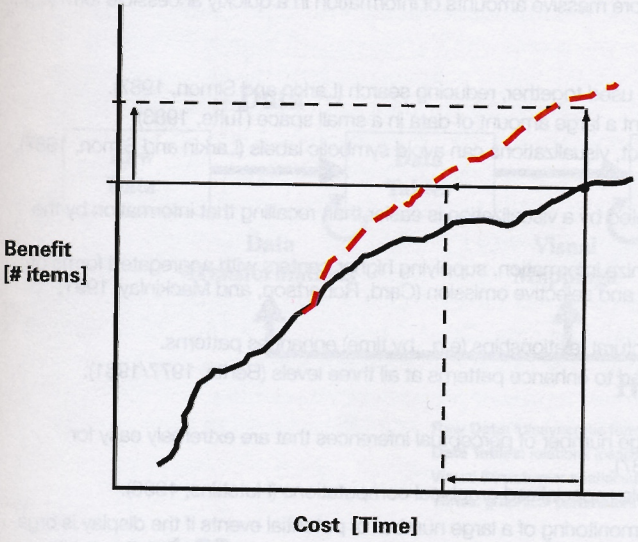


FIGURE **1.20**

Cost-of-Knowledge Characteristic Function.

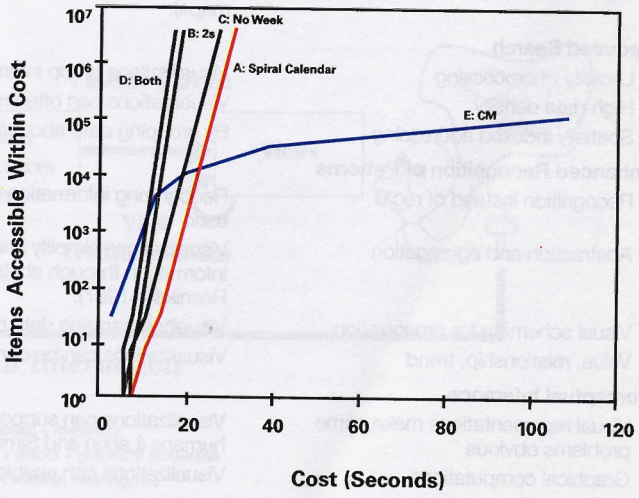


FIGURE **1.22**

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 attention 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 aerial.
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 mathematical 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*:

$$\{<\text{Value}_{ix}, \text{Value}_{iy}, \ldots>, <\text{Value}_{jx}, \text{Value}_{jy}, \ldots>, \ldots\}.$$

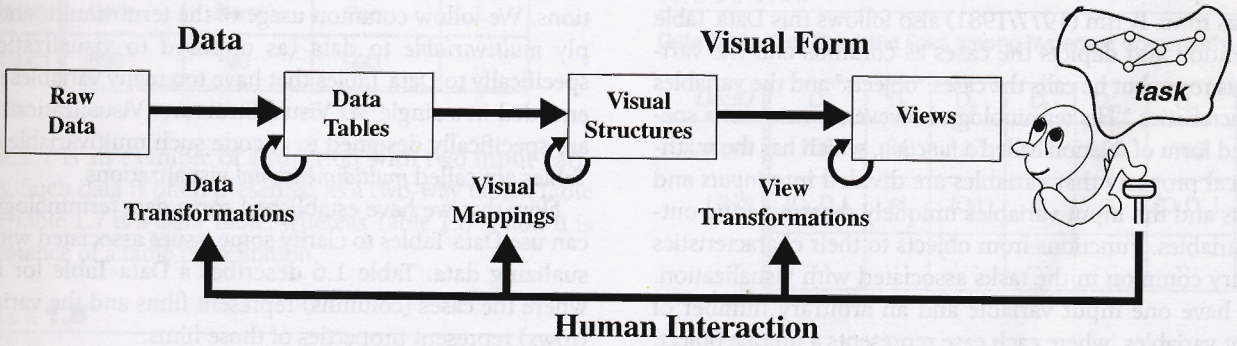


FIGURE **1.23**

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.

	$Case_i$	$Case_j$	$Case_k$	
$Variable_x$	$Value_{ix}$	$Value_{jx}$	$Value_{kx}$	...
$Variable_y$	$Value_{iy}$	$Value_{jy}$	$Value_{ky}$	...
...	...	...	...	...

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:

$$f(Case_i) = \langle Value_{ix}, Value_{iy}, \ldots \rangle.$$

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. $Case_i$ represents a unique object and the corresponding values represent the characteristics of that object.

Case	$Case_i$	$Case_j$	$Case_k$	...
$Variable_x$	$Value_{ix}$	$Value_{jx}$	$Value_{kx}$	...
$Variable_y$	$Value_{iy}$	$Value_{jy}$	$Value_{ky}$	...
...	...	...	...	...

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.

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

Start City	Basel	Basel	Berlin	...
End City	Berlin	Bern	Bern	...
Distance	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.

	Basel	**Berlin**	**Bern**	...
Basel	0	860	90	...
Berlin	860	0	930	...
Bern	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.

City	Basel	Berlin	Bern	...
Latitude	47.33N	52.32N	46.57N	...
Longitude	7.36E	13.25E	7.26E	...
Country	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.

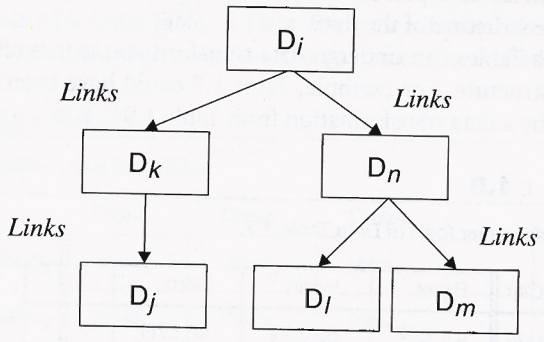
Data Tables can describe hierarchical and network data. To do this, a variable is used to describe the links between cases. For example, in Table 1.10 the variable *Links* describes the relationship among hypertext documents:

TABLE **1.10**

Data Table describing the links among hypertext documents.

DocID	D_i	D_j	D_k	D_l	D_m	D_n
Length	235	54	127	341	102	186
Links	$\{D_k D_n\}$	$\varnothing$	$\{D_j\}$	$\varnothing$	$\varnothing$	$\{D_l D_m\}$
...	...	...	...	...	...	...

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 ∅) 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 ≠ 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)

<Short, Medium, Long>.

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

{Goldfinger, Ben Hur, Star Wars}

can be sorted lexicographically

<Ben Hur, Goldfinger, Star Wars>.

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

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

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

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

for spatial variables that are specifically geophysical coordinates.

Other important subtypes are the temporal variables

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

and

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

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, Age20-35, 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.

FilmID	N	230	105	...
Title	N	Goldfinger	Ben Hur	...
Director	N	Hamilton	Wyler	...
Actor	N	Connery	Heston	...
Actress	N	Blackman	Harareet	...
Year	Q_t	1964	1959	...
Length	Q	112	212	...
Popularity	Q	7.7	8.2	...
Rating	O	PG	G	...
Film Type	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	*Mean*	*Sort* *Class* *Promote*
Structure	*Demote*	$X,Y,Z \rightarrow P_{xyz}$

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.

Individual	I1	I2	I3	I4	I5	I6	I7	I8	...
Ages	55	18	22	51	34	50	28	17	...
Income	1	6	8	10	4	7	3	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	...

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.

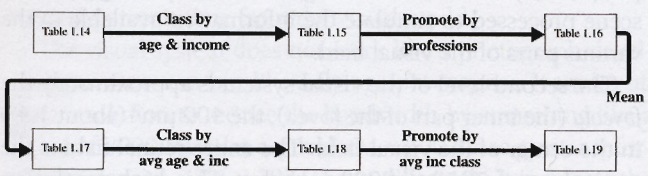


FIGURE **1.24**

Cascaded data transformations.

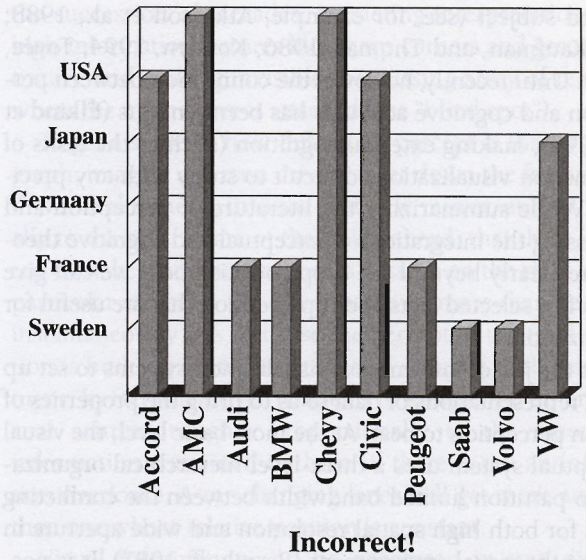


FIGURE **1.25**

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

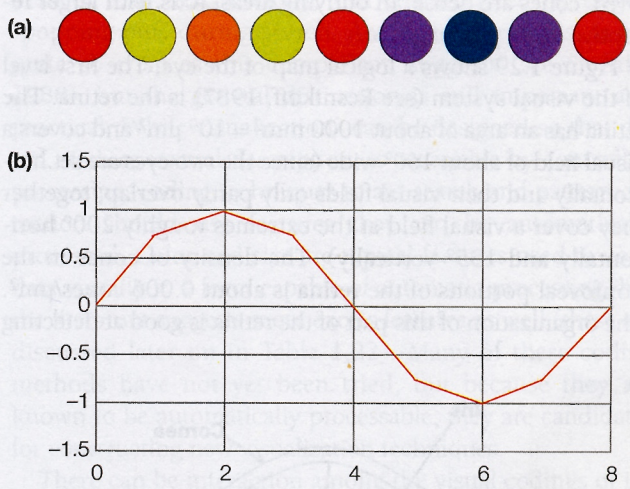


FIGURE **1.26**

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 mm² = 10^9 μm² and covers a visual field of about 160° 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° horizontally and 135° vertically). The density of cones in the nonfoveal portions of the retina is about 0.006 cones/μm². The organization of this part of the retina is good at detecting

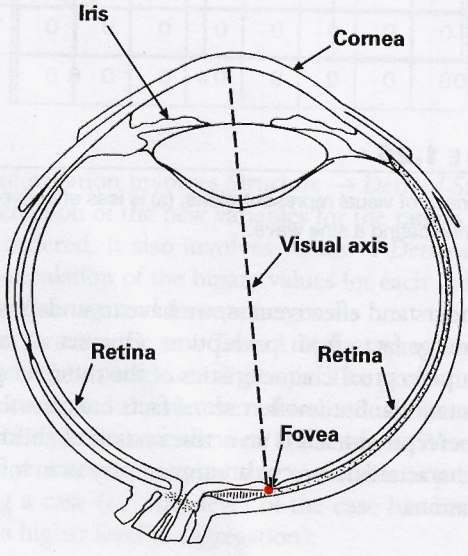


FIGURE **1.27**

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

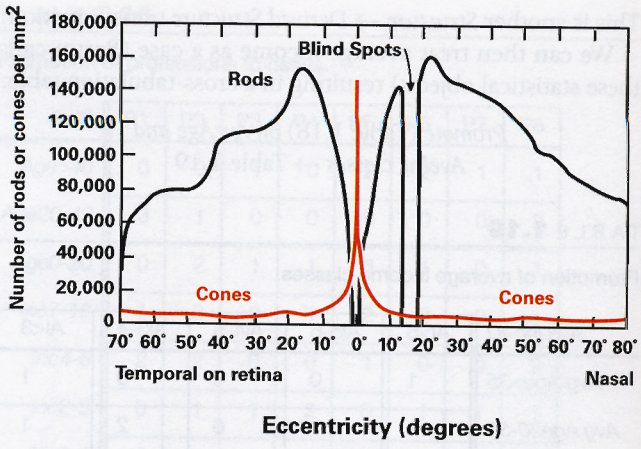


FIGURE **1.28**

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

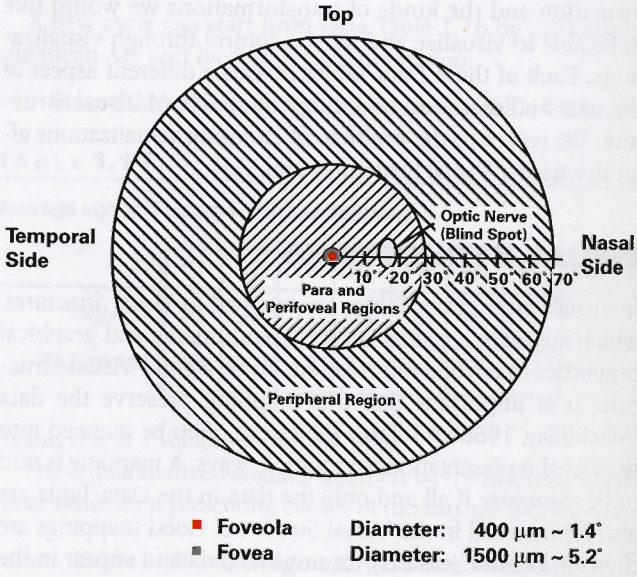


FIGURE **1.29**

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 μm² (about 1.4°) in the center of the visual field. The entire retinal field is the equivalent of 7950 ≈ 8000 foveolae. This high-resolution field is moved to points of interest about 1 to 5 times/sec at rates of up to 500°/sec, during which vision is suppressed.

(a) (b)

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 × 200 times greater information density = 1.6×10^9 (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 Tovée (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:

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

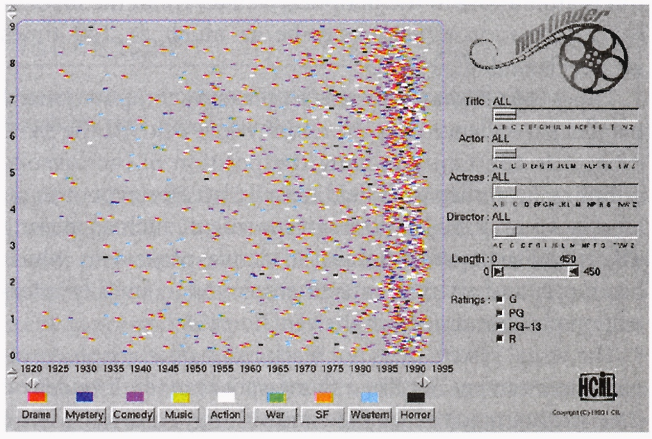


FIGURE **1.31**

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

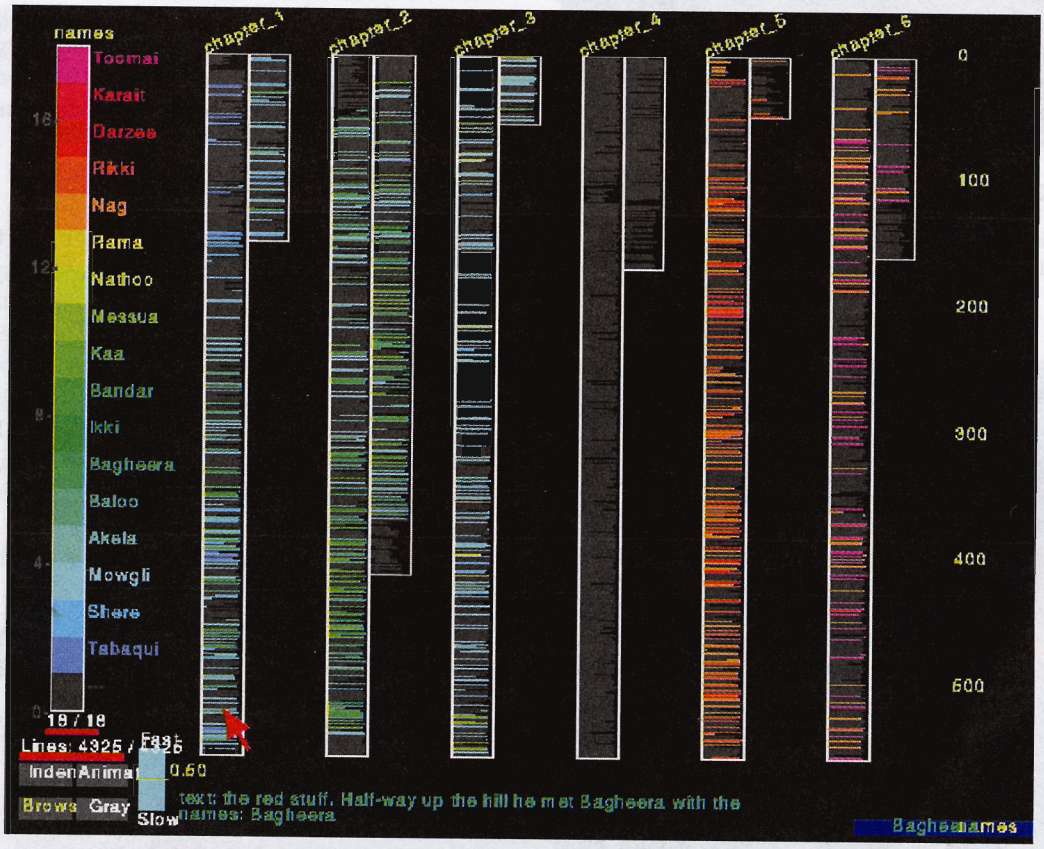


FIGURE **1.32**

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 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.

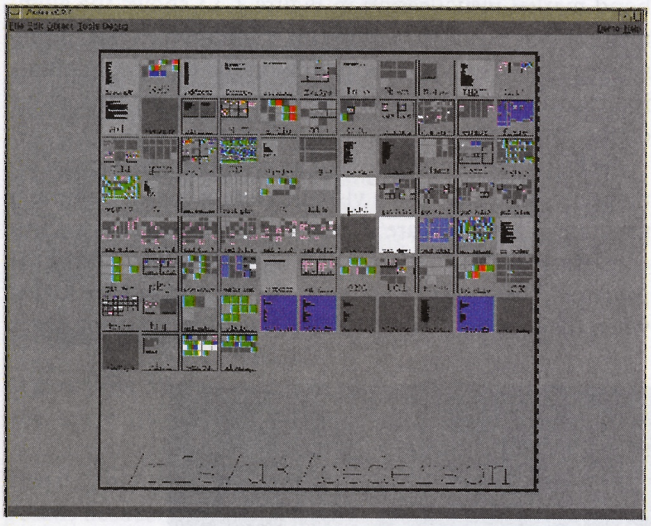


FIGURE **1.33**

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

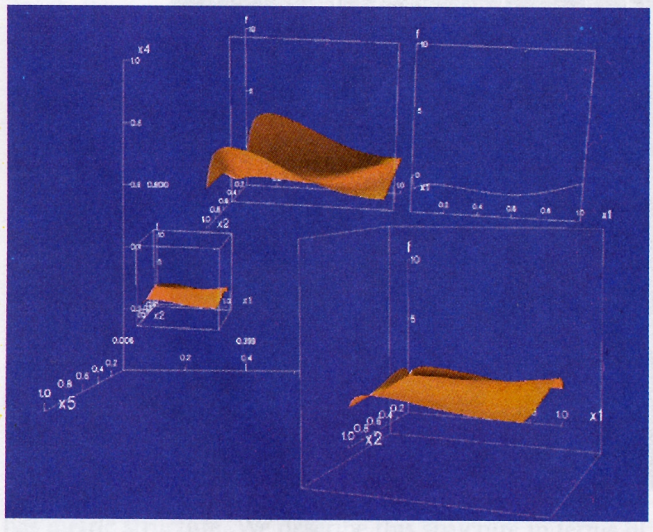


FIGURE **1.34**

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

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.

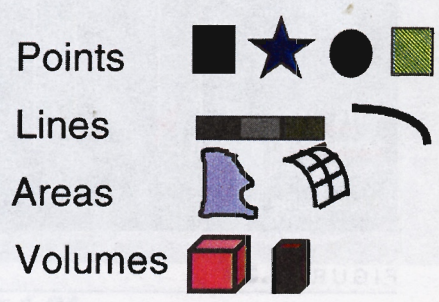


FIGURE **1.35**

Types of marks.

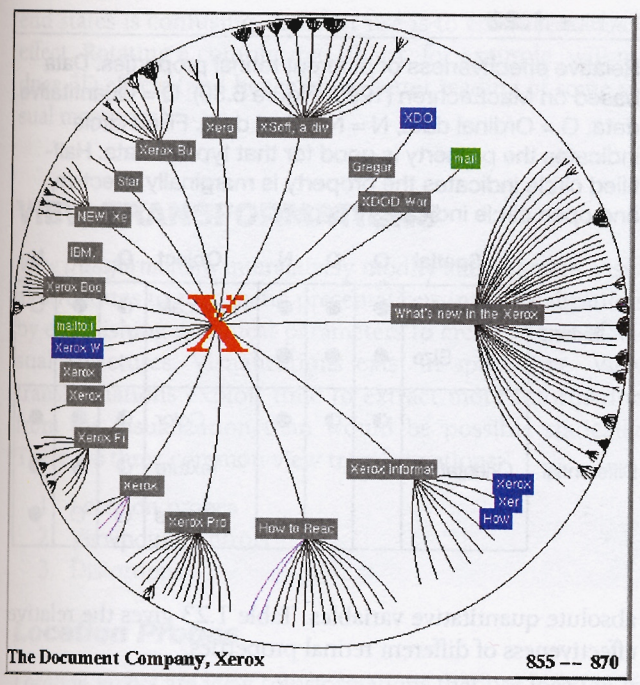


FIGURE **1.36**

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:

$$Links \rightarrow \text{Enclosure}.$$

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 FilmFinder in Figure 1.31 uses color to encode information in the scatterplot:

$$FilmID(FilmType) \rightarrow \text{P(Color)}$$

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

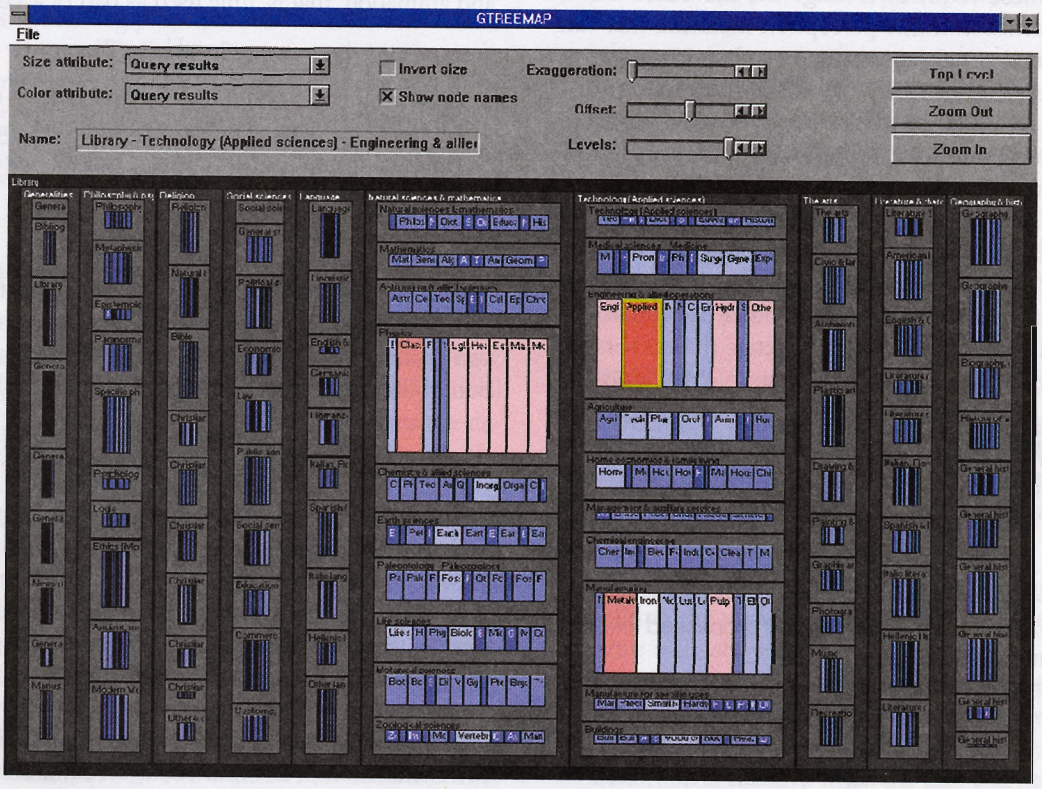


FIGURE **1.37**

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	Orientation	◑	◑	●	Color	◑	◑	●
					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:

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

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-

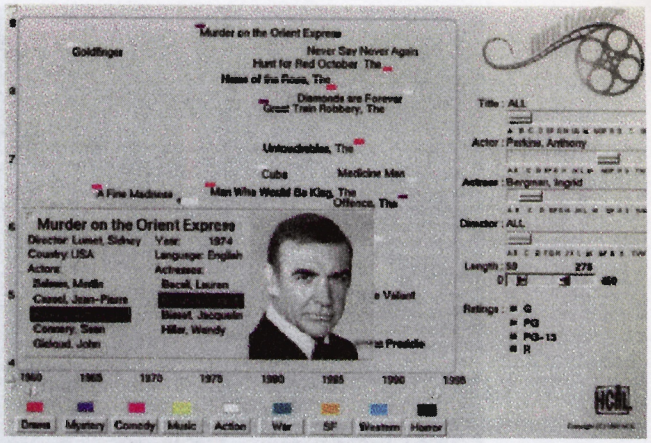


FIGURE **1.38**

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

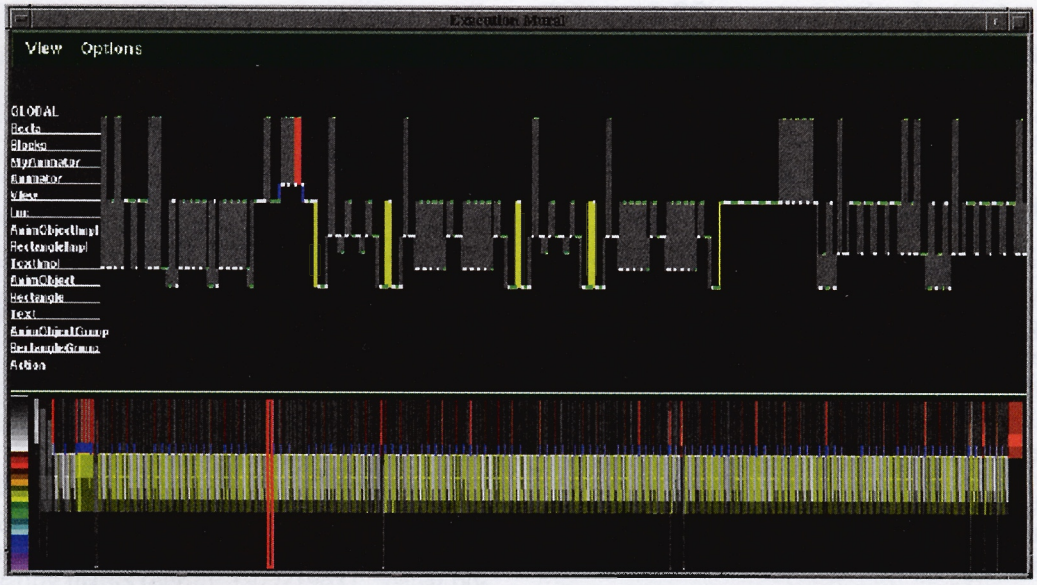


FIGURE **1.39**

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.

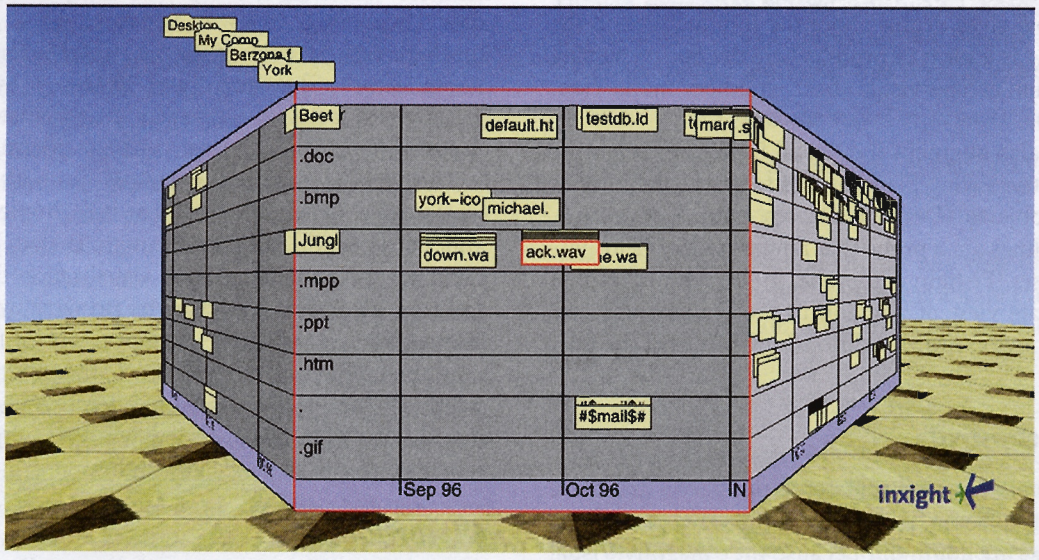


FIGURE **1.40**

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 Alighment 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 misadvised 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.