•26•

INFORMATION VISUALIZATION

Stuart Card
Xerox PARC

INTRODUCTION

The working mind is greatly leveraged by interaction with the world outside it. A conversation to share information, a grocery list to aid memory, a pocket calculator to compute square roots—all effectively augment a cognitive ability otherwise severely constrained by what is in its limited knowledge, by limited attention, and by limitations on reasoning. But the most profound leverage on cognitive ability is the ability to invent new representations, procedures, or devices that augment cognition far beyond its unaided biological endowment—and bootstrap these into even more potent inventions.

This chapter is about one class of inventions for augmenting cognition, collectively called "information visualization." Other senses could be employed in this pursuit—audition, for example, or a multi-modal combination of senses—the broader topic is really *information perceptualization*; however, in this chapter, we restrict ourselves to visualization. Visualization employs the sense with the most information capacity; recent advances in graphically agile computers have opened opportunities to exploit this capacity, and many visualization techniques have now been developed. A few examples suggest the possibilities.

Example 1: Finding Videos with the FilmFinder

The use of information visualization for finding things is illustrated by the FilmFinder (Ahlberg & Shneiderman, 1994a, 1994b). Unlike typical movie-finder systems, the FilmFinder is organized not around searching with keywords, but rather around rapid browsing and reacting to collections of films in the database Figure 26.1 shows a scattergraph of 2000 movies, plotting rated quality of the movie as a function of year when it was released. Color differentiates type of movies—comedy from drama and the like. The display provides an overview, the entire universe of all the movies, and some general features of the collection. It is visually apparent, for example, that a good share of the movies in the collection were released after 1965, but also that there are movies going back as far as the 1920s. Now the viewer "drills down" into the collection by using the sliders in the interface to show only movies with Sean Connery that are between 1 and 4½ hours in length (Fig. 26.2). As the sliders are moved, the display zooms in to show about 20 movies. It can be seen that these movies were made between 1960 and 1995, and all have a quality rating higher than 4. Since there is now room on the display, titles of the movies appear. Experimentation with the slider shows that restricting maximum length to 2 hours cuts out few interesting movies. The viewer chooses the highly rated movie, "Murder on the Orient Express" by double-clicking on its marker. Up pop details in a box (Fig. 26.3) giving names of other actors in the movie and more information. The viewer is interested in whether two of these actors, Anthony Perkins and Ingrid Bergman, have appeared together in any other movies. The viewer selects their names in the box, and then requests another search (Fig. 26.4). The result is a new display of two movies. In addition to the movie the viewer knew about, there is one other movie, a drama entitled "Goodbye,

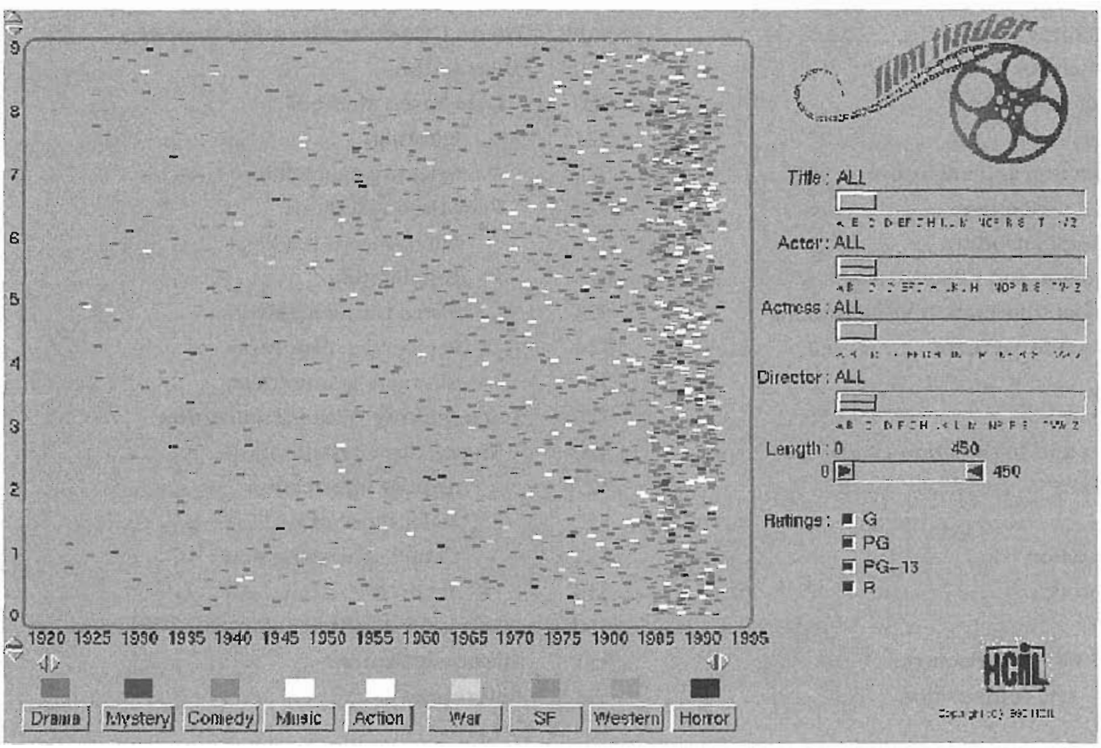


FIGURE 26.1. FilmFinder overview scattergraph. Courtesy University of Maryland.

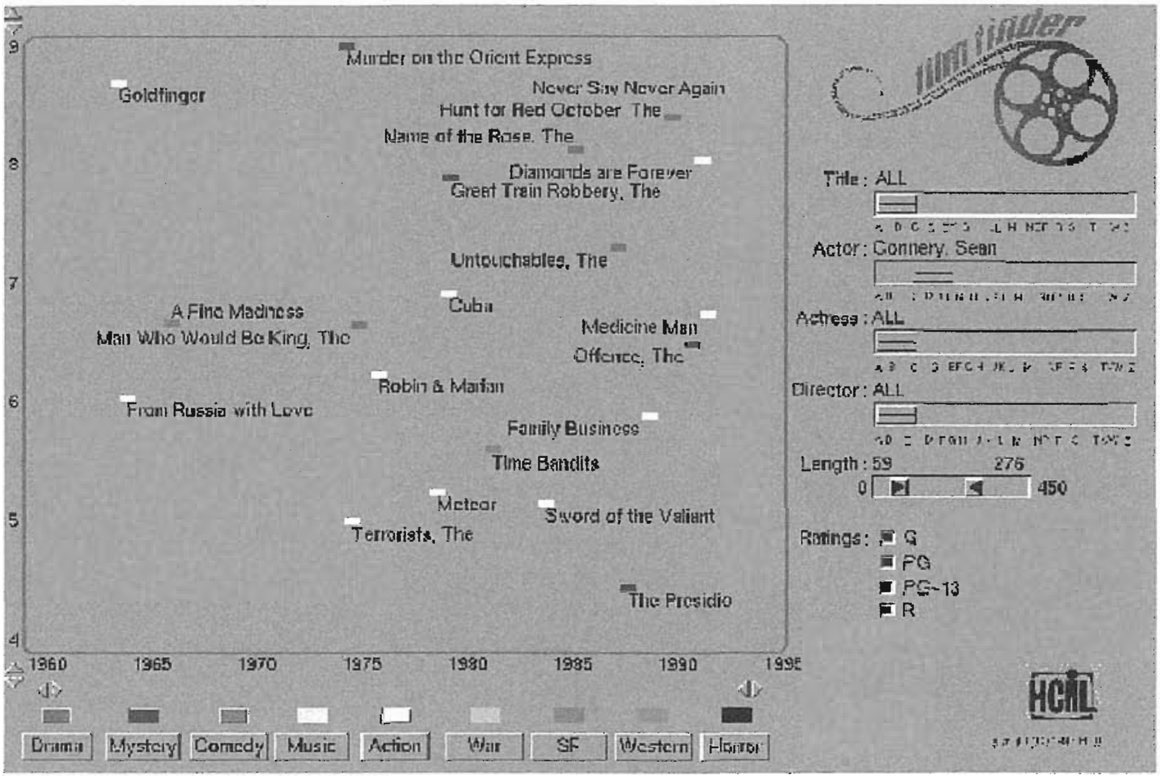


FIGURE 26.2. FilmFinder scattergraph zoom-in. Courtesy University of Maryland.

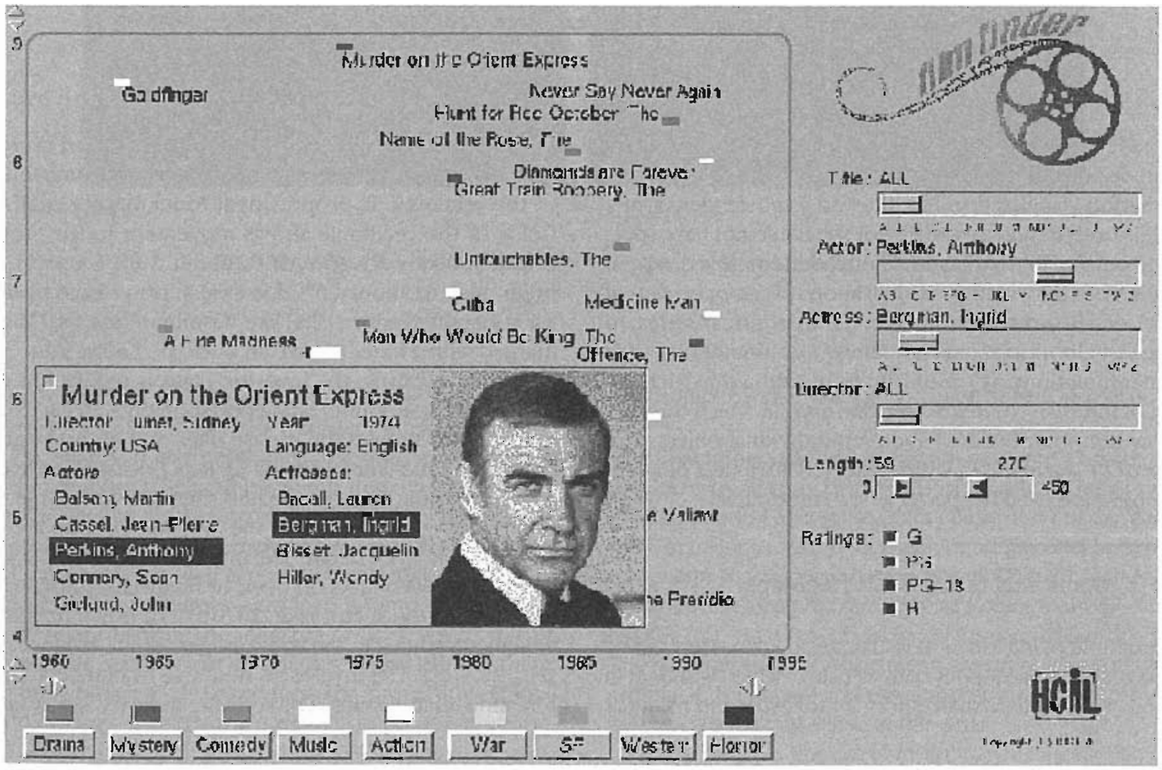


FIGURE 26.3. FilmFinder details on demand. Courtesy University of Maryland.

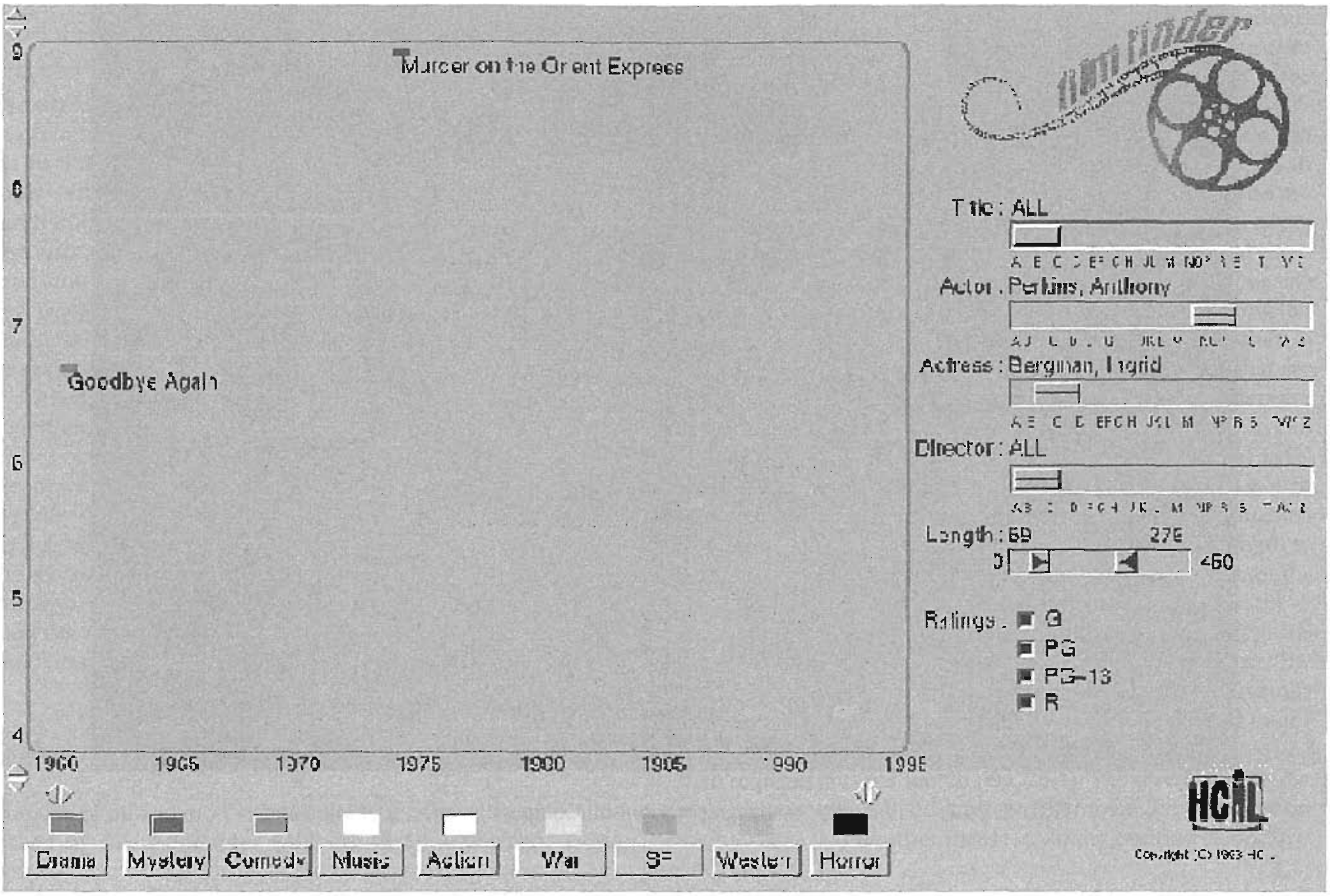


FIGURE 26.4. FilmFinder retrieval by example. Courtesy University of Maryland.

Again," made around 1960. The viewer is curious about this movie and decides to watch it.

Information visualization has allowed a movie viewer in a matter of seconds to find a movie he or she could not have specified at the outset. To do this, the FilmFinder employed several techniques from information visualization: (a) an *overview* of the collection showing its structure; (b) *dynamic queries*, in which the visualization seems to change instantaneously with control manipulations; (c) *zooming in* by adding restrictions to the set of interest; (d) *details on demand*, in which the user can display temporarily detail about an individual object, and (e) *retrieval by example*, in which selected attributes of an individual object are used to specify a new retrieval set.

Example 2: Monitoring Stocks with TreeMaps

Another example of information visualization is the TreeMap visualization on the SmartMoney.com website,[1] which is shown in Fig. 26.5(a). Using this visualization, an investor can monitor more than 500 stocks at once, with data updated every 15 minutes. Each colored rectangle in the figure is a company. The size of the rectangle is proportional to its market capitalization. Color of the rectangle shows movement in the stock price. Bright yellow corresponds to about a 6% increase in price, bright blue to about a 6% decrease in price. Each business sector is identified with a label like "Communications." Those items marked with a letter *N* have an associated news item.

In this example, the investor's task is to monitor the day's market and notice interesting developments. In Fig. 26.5(a), the investor has moved the mouse over one of the bright yellow rectangles, and a box identifying it as Erickson, with a +9.28% gain for the day, has popped up together with other information. Clicking on a box gives the investor a popup menu for selecting even more detail. The investor can either click to go to World Wide Web links on news or financials, or drill down, for example, to the sector (Fig. 26.5[b]), or down further to individual companies in the software part of the technology sector (Fig. 26.5[c]). The investor is now able to immediately note interesting relationships. The software industry is now larger than

[1]www.smartmoney.com

(a) (b) (c)

FIGURE 26.5. TreeMap of daily stock prices. Courtesy SmartMoney.com.

the hardware industry, for example, and despite a recent battering at the time of this figure, the Internet industry is also relatively large. Microsoft is larger than all the other companies in its industry combined. Selecting a menu item to look at year-to-date gains (Fig. 26.6), the investor immediately notes interesting patterns: Microsoft stock shows substantial gains, whereas Oracle is down; Dell is up, but Compaq is down; Tiny Advanced Micro is up, whereas giant Intel is neutral. Having noticed these relationships, the investor drills down to put up charts or analysts positions for companies whose gains in themselves, *or in relation to a competitor*, are interesting. For example, the investor is preparing a report on the computer industry for colleagues and notices how AMD is making gains against Intel, or how competition for the Internet is turning into a battle between Microsoft and AOL/Time Warner.

Example 3: Sensemaking with Permutation Matrices

As a final information visualization example, consider the case proposed by Bertin (1977/1981) of a hotel manager who wants to analyze hotel occupancy data (Table 26.1) to increase her return. In order to search for meaningful patterns in her data, she represents it as a permutation matrix (Fig. 26.7[a]. A permutation matrix is a graphic rendition of a cases x variables display. In Fig. 26.7(a), each cell of Table 26.1 is a small bar of a bar chart. The bars for cells below the mean are white; those above the bar are black. By permuting rows and columns, patterns emerge that lead to making sense of the data.

In Fig. 26.7(a), the set of months, which form the cases, are repeated to reveal periodic patterns across the end of the cycle. By visually comparing the pairs of rows, one can find rows

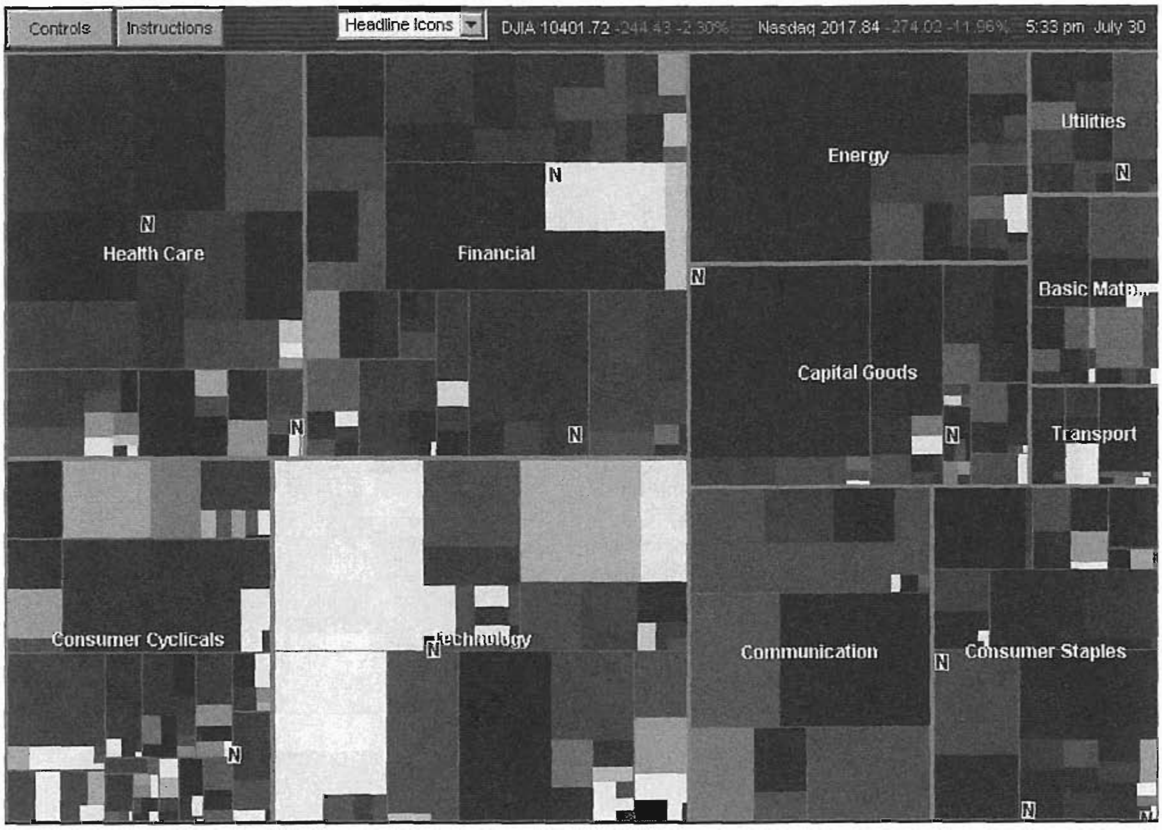


FIGURE 26.6. TreeMap of year-to-date stock prices. Courtesy SmartMoney.com.

TABLE 26.1. Data for Hotel Occupancy (Based on Bertin (1977/1981))

ID	VARIABLE	JAN	FEB	MAR	APR	MAY	JUNE	JULY	AUG	SEPT	OCT	NOV	DEC
1	% Female	26	21	26	28	20	20	20	20	20	40	15	40
2	% Local	69	70	77	71	37	36	39	39	55	60	68	72
3	% USA	7	6	3	6	23	14	19	14	9	6	8	8
4	% South America	0	0	0	0	8	6	6	4	2	12	0	0
5	% Europe	20	15	14	15	23	27	22	30	27	19	19	17
6	% M.East/Africa	1	0	0	8	6	4	6	4	2	1	0	1
7	% Asia	3	10	6	0	3	13	8	9	5	2	5	2
8	% *Businessmen*	78	80	85	86	85	87	70	76	87	85	87	80
9	% Tourists	22	20	15	14	15	13	30	24	13	15	13	20
10	% Direct Reservations	70	70	75	74	69	68	74	75	68	68	64	75
11	% Agency Reservations	20	18	19	17	27	27	19	19	26	27	21	15
12	% Air Crews	10	12	6	9	4	5	7	6	6	5	15	10
13	% Under 20	2	2	4	2	2	1	1	2	2	4	2	5
14	% 20–35	25	27	37	35	25	25	27	28	24	30	24	30
15	% 35–55	48	49	42	48	54	55	53	51	55	46	55	43
16	% Over 55	25	22	17	15	19	19	19	19	19	20	19	22
17	Price of rooms	163	167	166	174	152	155	145	170	157	174	165	156
18	*Length of stay*	1.7	1.7	1.7	1.91	1.9	2	1.54	1.6	1.73	1.82	1.66	1.44
19	% Occupancy	67	82	70	83	74	77	56	62	90	92	78	55
20	Conventions	0	0	0	1	1	1	0	0	1	1	1	1

that are similar. These are reordered and grouped (Fig. 26.7[b]). By this means, it is discovered that there seem to be two patterns of yearly variation. One pattern in Fig. 26.7(b) is semiannual, dividing the year into the cold months of October through April and the warm months of May through September. The other pattern breaks the year into four distinct regions. We have thus found the beginnings of a *schema*—that is, a framework in terms of which we can encode the raw data and describe it in a more compact language. Instead of talking about the events of the year in terms of individual months, we can now talk in terms of two series of periods, the semiannual one, and the four distinct periods. As we do so, there is a *residue* of information not included as part of our descriptive language. Sensemaking proceeds by the *omission and recoding of information into more compact form* (see Resnikoff, 1989). This residue of information may be reduced by finding a better or more articulated schema, or it may be left as noise. Beyond finding the basic patterns in the data, the hotel manager wants to make sense of the data relative to a purpose: she wants to increase the occupancy of the hotel. Therefore, she has also permuted general indicators of activity in Fig. 26.7(b), such as % Occupancy and Length of Stay, to the top of the diagram and put the rows that correlate with these below them. This reveals that Conventions, Businessmen, and Agency Reservations, all of which generally have to do with convention business, are associated with higher occupancy. This insight comes from the match in patterns *internal* to the visualization; it also comes from noting why these variables might correlate as a consequence of factors *external* to the visualization. She also discovers that marked differences exist between the winter and summer guests during the slow periods. In winter, there are more local guests, women, and age differences. In summer, there are more foreign tourists and less variation in age.

This visualization was useful for sensemaking on hotel occupancy data, but it is too complicated to communicate the high points. The hotel manager therefore creates a simplified diagram, Fig. 26.7(c). By graying some of the bars, the main points are more readily graspable, while still preserving the data relations. A December convention, for example, does not seem to have the effect of the other conventions in bringing in guests. It is shown in gray as residue in the pattern. The hotel manager suggests moving the convention to another month, where it might have more effect on increasing the occupancy of the hotel.

What Is Information Visualization?

The FilmFinder, the TreeMap, and the permutation matrix hotel analysis are all examples of the use of information visualization. We can define information visualization as "the use of computer-supported, interactive, visual representations of abstract data in order to amplify cognition" (Card, Mackinlay, & Shneiderman, 1999).

Information visualization needs to be distinguished from related areas: *scientific visualization* is like information visualization, but it is applied to scientific data and typically is physically based. The starting point of a natural geometrical substrate for the data, whether the human body or earth geography, tends to emphasize finding a way to make visible the invisible (say, velocity of air flow) within an existing spatial framework. The chief problem for information visualization, in contrast, is often finding an effective mapping between abstract entities and a spatial representation. Both information visualization and scientific visualization belong to the broader field of *data graphics*, which is the use of abstract, nonrepresentational visual representations to amplify cognition. Data graphics, in turn, is part of *information design*, which concerns itself with external representations for amplifying cognition. At the highest level, we could consider information design a part of *external cognition*, the uses of the external world to accomplish some cognitive process. Characterizing the purpose of information visualization as *amplifying cognition* is purposely broad. Cognition can be the process of writing a scientific paper or shopping on the Internet for a cell phone. Generally, it refers to the intellectual processes in which information is obtained, transformed, stored, retrieved, and used. All of these can be advanced generally by means of external cognition, and in particular by means of information visualization.

Why Does Visualization Work?

Visualization aids cognition not because of some mystical superiority of pictures over other forms of thought and communication, but rather because visualization helps the user by making the world outside the mind a resource for thought in specific ways. We list six groups of these in Table 26.2 (Card et al., 1999): Visualization amplifies cognition by (a) increasing the memory and processing resources available to the users, (b) reducing search for information, (c) using visual representations to enhance the detection of patterns, (d) enabling perceptual inference operations, (e) using perceptual attention mechanisms for monitoring, and (f) by encoding information in a manipulable medium. The FilmFinder, for example, allows the representation of a large amount of data in a small space in a way that allows patterns to be perceived visually in the data. Most important, the method of instantly responding in the display to the dynamic movement of the sliders allowed users to rapidly explore the multidimensional space of films. The TreeMap of the stock market allows monitoring and exploration of many equities. Again, much data is represented in little space. In this case, the display manages the user's attention, drawing it to those equities with unusually large changes, and supplying the means to drill down into the data to understand why these movements may be happening. In the hotel management case, the visual representation makes it easier to notice similarities of behavior in a multidimensional attribute space, then to cluster and rerepresent these. The final product is a compact (and simplified) representation of the original data that supports a set of forward decisions. In all of these cases, visualization allows the user to (a) examine a large amount of information, (b) keep an overview of the whole while pursuing details, (c) keep track of (by using the display as an external working memory) many things, and (d) produce an abstract representation of a situation through the omission and recoding of information.

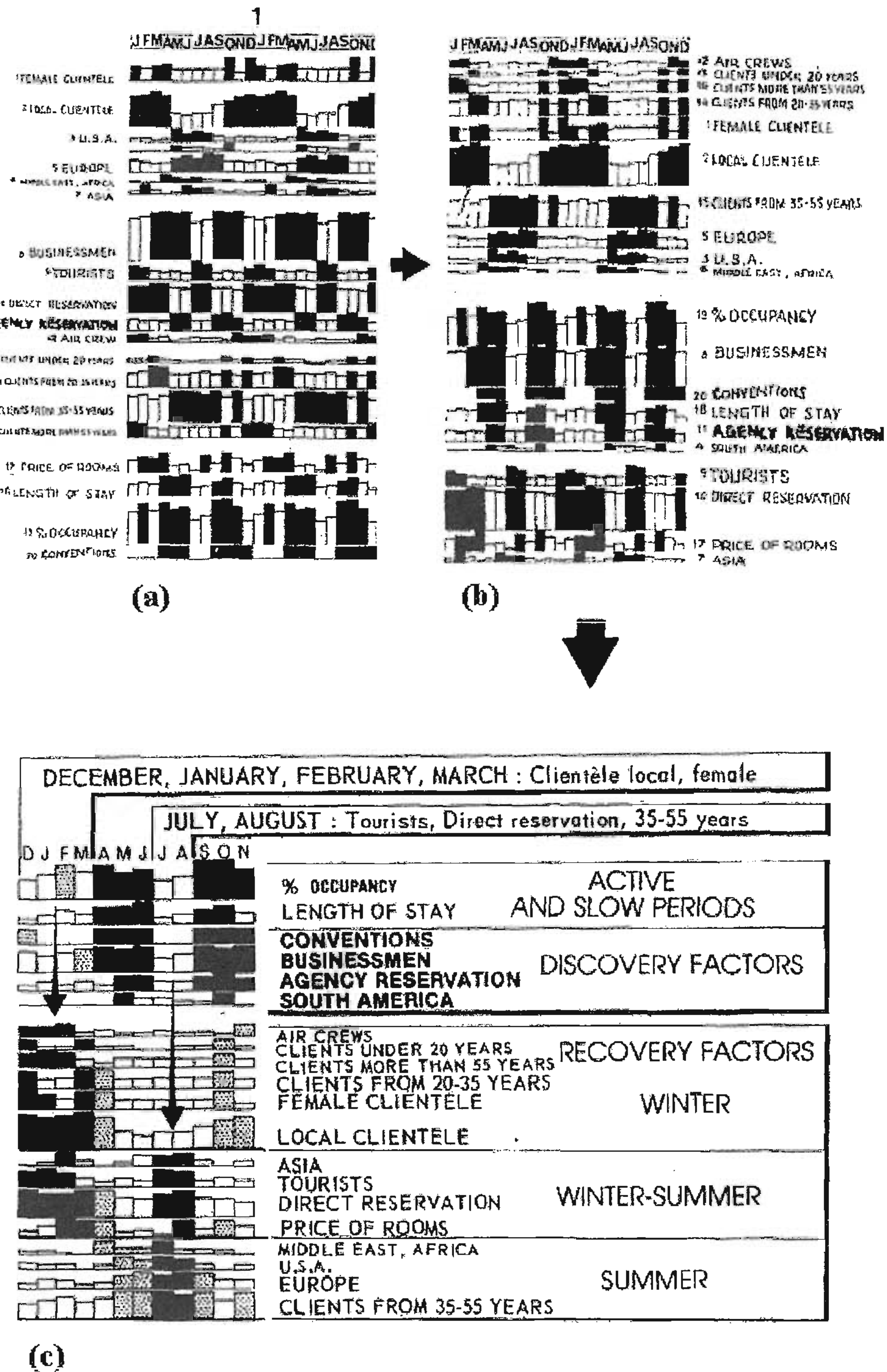


FIGURE 26.7. Permutation matrix representation of hotel data from (Berlin, 1977/1981). (a) Initial matrix of variables. (b) Permuted matrix to group like patterns together.

Historical Origins

Drawn visual representations have a long history. Maps go back millennia. Diagrams were an important part of Euclid's books on geometry. Science, from earliest times, used diagrams to (a) record observations, (b) induct relationships, (c) explicate methodology of experiments, and (d) classify and conceptualize phenomena (for a discussion, see Robin, 1992). For example, Fig. 26.8 is a hand-drawn illustration, in Newton's first scientific publication, illustrating how white light is really composed of

TABLE 26.2. How Information Visualization Amplifies Cognition

1. Increased Resources	
High-bandwidth hierarchical interaction	Human moving gaze system partitions limited channel capacity so that it combines high spatial resolution and wide aperture in sensing the visual environments (Larkin & Simon, 1987).
Parallel perceptual processing	Some attributes of visualizations can be processed in parallel compared to text, which is serial.
Offload work from cognitive to perceptual system	Some cognitive inferences done symbolically can be recoded into inferences done with simple perceptual operations (Larkin & 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).
2. Reduced Search	
Locality of processing.	Visualizations group information used together reducing search (Larkin & 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 & Simon, 1987).
3. 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, & Mackinlay, 1991); (Resnikoff, 1989).
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, 1967/1983).
4. Perceptual Inference	
Visual representations make some problems obvious	Visualizations can support a large number of perceptual inferences that are very easy for humans (Larkin & Simon, 1987).
Graphical computations	Visualizations can enable complex specialized graphical computations (Hutchins, 1996).
5. 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.
6. Manipulable medium	Unlike static diagrams, visualizations can allow exploration of a space of parameter values and can amplify user operations.

Source: Card, Mackinlay, & Shneiderman, 1999.

many colors. Sunlight enters from the window at right and is refracted into many colors by a prism. One of these colors can be selected (by an aperture in a screen) and further refracted by another prism, but the light stays the same color, showing that it has already been reduced to its elementary components. As in Newton's illustration, early scientific and mathematical diagrams generally had a spatial, physical basis and were used to reveal the hidden, underlying order in that world.

Surprisingly, diagrams of abstract, nonphysical information are apparently rather recent. Tufte (1983) dates abstract diagrams to (Playfair, 1786) in the 18th century. Figure 26.9 is one of Playfair's earliest diagrams. The purpose was to convince readers that English imports were catching up with imports. Starting with Playfair, the classical methods of plotting data were developed—graphs, bar charts, and the rest.

Recent advances in the visual representation of abstract information derive from several strands that became intertwined. In 1967, Bertin (1967/1983, 1977/1981), a French cartographer published his theory of *The Semiology of Graphics.* This theory identified the basic elements of diagrams and their combination. Tufte (1983, 1990, 1997), from the fields of visual design and data graphics, published a series of seminal books that set forth principles for the design of data graphics and emphasized maximizing the density of useful information. Both Bertin's and Tufte's theories became well known and influential. Meanwhile, within statistics, Tukey (1977) began a movement on exploratory data analysis. His emphasis was not on the quality of graphical presentation, but on the use of pictures to give rapid, statistical insight into data relations. For example, "box and whisker plots" allowed an analyst to get a rapid characterization of data distributions. Cleveland and McGill (1988) wrote an influential book, *Dynamic Graphics for Statistics*, explicating new visualizations of data with particular emphasis on the visualization of multidimensional data.

In 1985, NSF launched an initiative on *scientific visualization* (McCormick & DeFanti, 1987). The purpose of this initiative was to use advances in computer graphics to create a new class of analytical instruments for scientific analysis, especially as a tool for comprehending large, newly produced datasets in the geophysical and biological sciences. Meanwhile, the computer graphics and artificial intelligence communities were interested in the automatic design of visual presentations of data. Mackinlay's (1986a, 1986b) thesis APT formalized Bertin's design theory, added psychophysical data, and used these to build a system for automatically generating diagrams of data, tailored for some purpose. 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. This community was interested not so much in the quality of the graphics as in the automation of the match between data characteristics, presentational purpose, and graphical presentation. Finally, the user interface community saw advances in graphics hardware opening the possibility of a new generation of user interfaces. The first use of the term "information visualization" was probably in

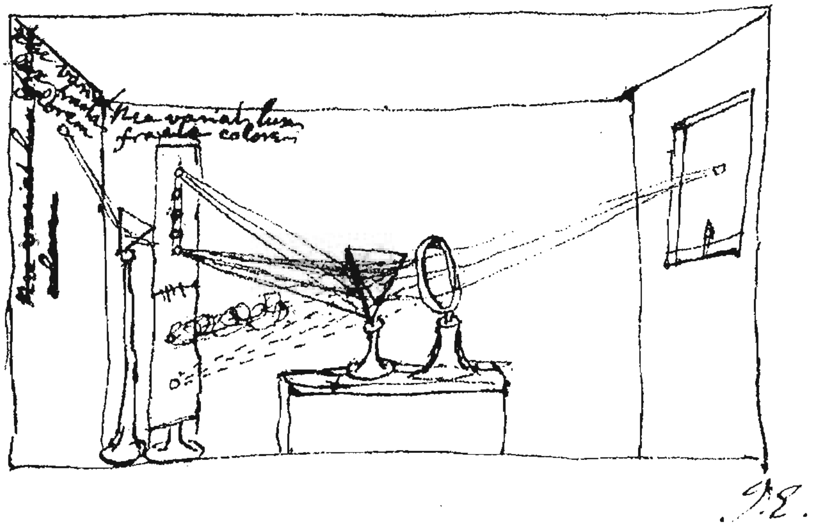


FIGURE 26.8. Newton's optics illustration (from Robin, 1992).

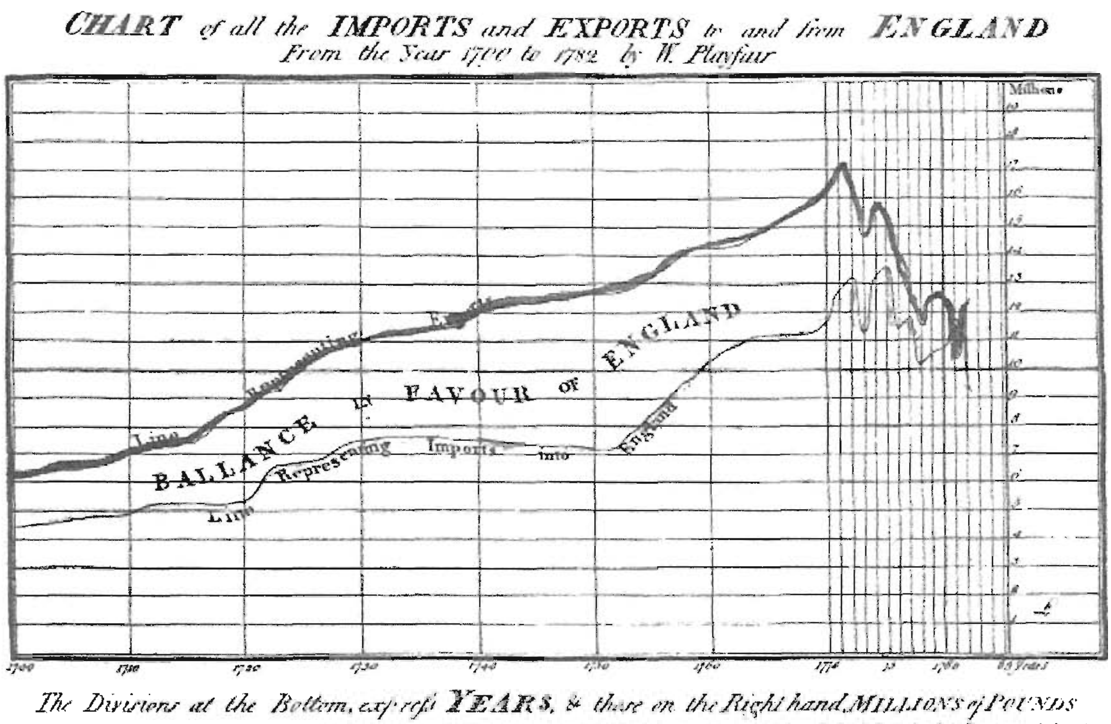


FIGURE 26.9. Playfair's charts of English imports and exports, from (Tufte, 1983).

Robertson, Card, and Mackinlay (1989). Early studies in this community focused on user interaction with large amounts of information: Feiner and Beshers (1990) presented a method, worlds within worlds, for showing six-dimensional financial data in an immersive virtual reality. Shneiderman (1992) developed a technique called "dynamic queries" for interactively selecting subsets of data items and TreeMaps, a space-filling representation for trees. Robertson, Card, and Mackinlay (1993) presented ways of using animation and distortion to interact with large data sets in a system called the Information Visualizer, which used *focus + context* displays to nonuniformly present large amounts of information. The emphasis for these studies was on the means for cognitive amplification, rather than on the quality of the graphics presentations.

The remainder of this chapter will concentrate on the techniques that have been developed for mapping abstract information to interactive visual form to aid some intellectual task. The perceptual foundations of this effort are beyond the scope of this chapter, but are covered in Ware (2000). Further details on information visualization techniques are addressed in a text by Spence (2000). The classic papers in information visualization are collected in Card et al. (1999).

THE VISUALIZATION REFERENCE MODEL

Mapping Data to Visual Form

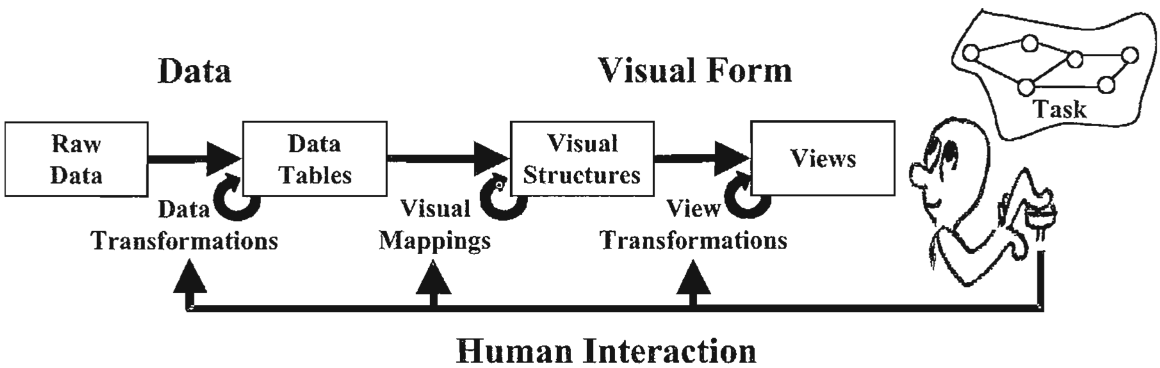
Despite their seeming variability, information visualizations can be systematically analyzed. Visualizations can be thought of as adjustable mappings from data to visual form to the human perceiver. In fact, we can draw a simple Visualization Reference Model of these mappings (Fig. 26.10). Arrows follow from *Raw Data* (data in some idiosyncratic format) on the left, though a set of *Data Transformations* into *Data Tables* (canonical descriptions of data in a variables x cases format extended to include metadata). The most important mapping is the arrow from Data Tables to *Visual Structures* (structures that combine values an available vocabulary of visual elements—spatial substrates, marks, and graphical properties). Visual Structures can be further transformed by *View Transformations*, such as visual distortion or 3D viewing angle, until it finally forms a *View* that can be perceived by human users. Thus, Raw Data might start out as text represented 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, in turn, might be reduced by multidimensional scaling to create the analytic abstraction to be visualized, expressed as a Data Table of x, y, z coordinates that could be displayed. These coordinates might be transformed into a Visual Structure—that is, a surface on an information landscape—which is then viewed at a certain angle.

Similar final effects can be achieved by transformations at different places in the model: When a point is deleted from the visualization, has the point been deleted from the dataset? Or is it still in the data merely not displayed? Chi and Riedl (1998) called this the *view-value distinction*, and it is an example of just one issue where identifying the locus of a transformation using the Visualization Reference Model helps to avoid confusion.

Information visualization is about the not just creation of visual images, but also the interaction with those images in the service of some problem. In the Visualization Reference Model, another set of arrows flow back from the human at the right into the transformations themselves, indicating the adjustment of these transformations by user-operated controls. It is the rapid reciprocal reaction between the generation of images by machine and the selection and parametric adjustment of those images, giving rise to new images that gives rise to the attractive power of interactive information visualization.

Data Structures

It is convenient to express Data Tables as tables of objects and their attributes, as in Table 26.3. For example, in the FilmFinder, the basic objects (or "cases") are films. Each film is associated with a number of attributes or variables, such as title, stars, year of release, genre type, and so forth. The vertical double black line in the table separates data in the table to the left of the line



Raw Data: idiosyncratic formats
Data Tables: relations (cases by variables) + meta-data
Visual Structures: spatial substrates + marks + graphical properties
Views: graphical parameters (position, scaling, clipping, ...)

FIGURE 26.10. Reference model for visualization (Card et al., 1999). Visualization can be described as the mapping of data to visual form that supports human interaction in a workplace for visual sense making.

TABLE 26.3. 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	...
FilmType	Action	Action	Drama	...

Source: (Card et al., 1999).

from the metadata, expressed as variable names, to the left of the line. The horizontal black line across the table separates input variables from output variables—that is, the table can be thought of as a function.

$$f(\text{input variables}) = \text{output variables}.$$

So,

$$Year\ (FilmID = 105) = 1959.$$

Variables imply a scale of measurement, and it is important to keep these straight. The most important to distinguish are

N = *Nominal (are only = or ≠ to other values)*
O = *Ordinal* (obeys a < relation)
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].

In addition to the three basic types of variables, subtypes represent important properties of the world associated with specialized visual conventions. We sometimes distinguish the subtype *Quantitative Spatial* (Q_s) for intrinsically spatial variables common in scientific visualization and the subtype *Quantitative Geographical* (Q_g) for spatial variables that are specifically geophysical coordinates. Other important subtypes are similarity metrics *Quantitative Similarity* (Q_m), and the temporal variables *Quantitative Time* (Q_t) and *Ordinal Time* (O_t). We can also distinguish Interval Scales (I) (like Quantitative Scales, but since there is not a natural zero point, it is not meaningful to take ratios). An example would be dates. It is meaningful to subtract two dates (June 5, 2002 – June 3, 2002 = 2 days), but it does not make sense to divide them (June 5, 2002 ÷ June 23, 2002 = Undefined). Finally, we can define an *Unstructured* Scale (U), whose only value is present or absent (e.g., an error flag). The scales are summarized in Table 26.4.

Scale types can be altered by transformations, and this practice is sometimes convenient. For example, quantitative variables can be mapped by data transformations into ordinal variables

$$Q \rightarrow O$$

by dividing them into ranges. For example, film lengths [0, 360] minutes (type Q) can be broken into the ranges (type O),

[0, 360] minutes → ⟨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. In the other direction, nominal variables can be transformed to ordinal values

$$N \rightarrow O$$

based on their name. For example, film titles {GOLDFINGER, BEN HUR, STAR WARS} can be sorted lexicographically

{GOLDFINGER, BEN HUR, STAR WARS} →
⟨BEN HUR, GOLDFINGER, STAR WARS⟩.

Strictly speaking, we have not transformed their values, but in many uses (e.g., building alphabetically arranged dictionaries of words or sliders in the FilmFinder), we can act as if we had.

Variable scale types form an important class of metadata that, as we shall see, is important for proper information visualization. We can add scale type to our Data Table in Table 26.3 together with cardinality or range of the data to give us essentially a codebook of variables as in Table 26.5.

Visual Structures

Information visualization maps data relations into visual form. At first, it might seem that a hopelessly open set of visual forms can result. Careful reflection, however, reveals what every artist knows: that visual form is subject to strong constraints. Visual form that reflects the systematic mapping of data relations onto visual form, as in information visualization or data graphics, is subject to even more constraints. It is a genuinely surprising fact, therefore, that most information visualization involves the mapping data relations onto only a half dozen components of visual encoding:

1. *Spatial substrate*
2. *Marks*
3. *Connection*
4. *Enclosure*
5. *Retinal properties, or*
6. *Temporal encoding*

Of these mappings, the most powerful is how data are mapped onto the spatial substrate—that is, how data are mapped into spatial position. In fact, one might say that the design of an information visualization consists first of deciding which variables are going to get the spatial mappings, and then how the rest of the variables are going to make do with the coding mappings that are left.

Spatial substrate. As we have just said, the most important choice in designing an information visualization is which variables are going to map onto spatial position. This decision

TABLE 26.4. Classes of Data and Visual Elements

	Data Classes		Visual Classes	
Class	Description	Example	Description	Example
U	*Unstructured* (can only distinguish presence or absence)	ErrorFlag	*Unstructured* (no axis, indicated merely whether something is present or absent)	Dot
N	*Nominal* (can only distinguish whether two values are equal)	{Goldfinger, Ben Hur, Star Wars}	*Nominal Grid* (a region is divided into subregions, in which something can be present or absent)	Colored circle
O	*Ordinal* (can distinguish whether one value is less or greater but not difference or ratio)	(Small, Medium, Large)	*Ordinal Grid* (order of the subregions is meaningful)	Alpha slider
I	*Interval* (can do subtraction on values, but no natural zero and can't compute ratios)	|10 Dec. 1978–4 Jun. 1982|	*Interval Grid* (region has a metric but no distinguished origin)	Year axis
Q	*Quantitative* (can do arithmetic on values)	|0–100| kg	*Quantitative Grid* (a region has a metric)	Time slider
Q_s	—*Spatial variables*	|0–20| m	—*Spatial grid*	
Q_m	—*Similarity*	|0–1|	—*Similarity space*	
Q_g	—*Geographical coord.*	|30°N–50°N|Lat.	—*Geographical cood.*	
Q_t	—*Time variable*	|10–20| μsec	—*Time grid*	

TABLE 26.5. Data Table with Meta-Data 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	...
FilmType	N	Action	Action	...

Source: (Card et al., 1999).

gives importance to spatially encoded variables at the expense of variables encoded using other mappings. Space is perceptually dominant (MacEachren, 1995); it is good for discriminating values and picking out patterns. It is easier, for example, to identify the difference between a sine and a tangent curve when encoded as a sequence of spatial positions than as a sequence of color hues.

Empty space itself, as a container, can be treated as if it had metric structure. Just as we classified variables according to their scale type, we can think of the properties of space in terms of the scale type of an axis of space (cf. Engelhardt, Bruin, Janssen, & Scha, 1996). Axis scale types correspond to the variable scale types (see Table 26.4). The most important axes are

U = Unstructured (no axis, indicated merely whether something is present or absent)
N = Nominal Grid (a region is divided into subregions, in which something can be present or absent)
O = Ordinal Grid (the ordering of these subregions is meaningful), and
Q = Quantitative Grid (a region has a metric).

Besides these, it is convenient to make additional distinctions for frequently used subtypes, such as Spatial axes (Qs).

Axes can be linear or radial; essentially, they can involve any of the various coordinate systems for describing space. Axes are an important building block for developing Visual Structures. Based on the Data Table for the FilmFinder in Table 26.5, we represent the scatterplot of as composed of two orthogonal quantitative axes:

$$\textit{Year} \rightarrow Q_x,$$
$$\textit{Popularity} \rightarrow Q_y.$$

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

$$\textit{Ratings} \rightarrow O_y.$$

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

$$\textit{FilmType} \rightarrow N_x.$$

Marks. Marks are the visible things that occur in space. There are four elementary types of marks (Fig. 26.11):

1. P = Points (0D),
2. L = Lines (1D),
3. A = Areas (2D), and
4. 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 such as shape.

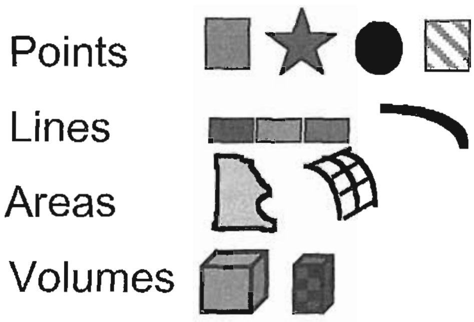


FIGURE 26.11. Types of marks.

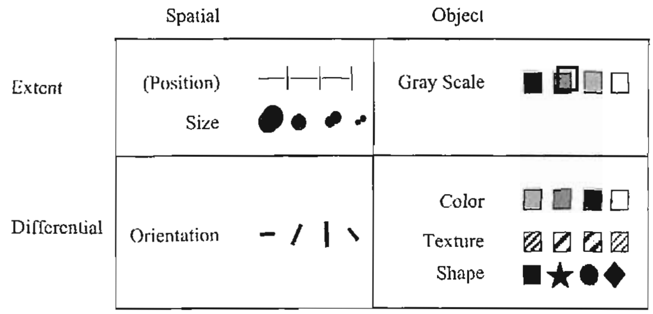


FIGURE 26.12. Retinal properties (Card et al., 1999). The six retinal properties can be grouped by whether they form a scale with a natural zero point (extend) and whether they deal with spatial distance or orientation (spatial).

Connection and enclosure. Point marks and line marks can be used to signify other sorts of topological structure: graphs and trees. These allow showing relations among objects without the geometrical constraints implicit in mapping variables onto spatial axes. Instead, we draw explicit lines. Hierarchies and other relationships can also be encoded using enclosure. Enclosing lines can be drawn around subsets of items. Enclosure can be used for trees, contour maps, and Venn Diagrams.

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 Fig. 26.1 uses color to encode information in the scatterplot:

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

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

Figure 26.12 shows Bertin's six "retinal variables" separated into spatial properties and object properties according to which area of the brain they are believed to be processed (Kosslyn, 1994). They are sorted 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. MacEachren (1995) has proposed (a) crispness (the inverse of the amount of distance used to blend two areas or a line into an area), (b) resolution (grain with raster or vector data will be displayed), (c) transparency, and (d) arrangement (e.g., different ways of configuring dots). He further proposed dividing color into (a) value (essentially, the gray level of Fig. 26.12), (b) hue, and (c) saturation. Graphical properties from the perception literature that can support preattentive processing have been suggested candidates for coding variables such as curvature, lighting direction, or direction of motion (see Healey, Booth, and Enns, 1995). All of these suggestions require further research.

Temporal encoding. Visual Structures can also temporally encode information; human perception is very sensitive to changes in mark position and the mark's retinal properties. We need to distinguish between temporal data variables to be visualized

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

and animation, that is, mapping a variable into time,

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

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 as data.

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

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),

$$Q_t \rightarrow Qx \textit{ [make time be the X-axis]}$$
$$f(Q_t) \rightarrow Qy, \textit{ [make accumulated rainfall be the Y-axis,}$$

then we can directly experience rates as visual linear slope, and we can experience changes in rates as curves. This encoding of time into space for display allows us to make much more precise judgments about rates than would be possible from encoding time as time. 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 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 (hence, allow better reading of some visual mappings).

Expressiveness and Effectiveness

Visual mappings transform Data Tables into Visual Structure and then into a visual image. This image is not just an arbitrary image. It is an image that has a particular meaning it must express. That meaning is the data relation of which it is the visual transformation. We can think of the image as a sentence in a visual language (Mackinlay, 1986b) that expresses the relations in the Data Table. To be a good information visualization, the mappings must satisfy some constraints. The first constraint is that the mapping must be expressive. A visualization is said to be *expressive* if and only if it encodes all the data relations intended and no other data relations. The first part of expressiveness turns out to be easier than the second. Suppose we plot FilmType against Year using the data-to-visual mapping in Fig. 26.13. The problem of this mapping is that the nominal movie rating data are expressed by a quantitative axis. That is, we have tried to map

$$FilmType(N) \rightarrow Position(Q).$$

In so doing, we have visually expressed all the data relation, but the visualization also implies relationships that do not exist. For example, the 1959 version of Ben Hur does not have a film type that is five times greater than the 1926 version of Ben Hur, as implied in the figure. Wisely, the authors of the FilmFinder chose the mapping

$$FilmType(N) \rightarrow Color(N).$$

Of course, there are circumstances in which color could be read as ordinal, or even possibly quantitative, but the miscellaneous order of the buttons in Fig. 26.1 discourages such an interpretation and the relatively low effectiveness of color for this purpose in Table 26.7 also discourages this interpretation.

Table 26.6 shows the mappings chosen by authors of the FilmFinder. The figure shows the Data Table's metadata and data

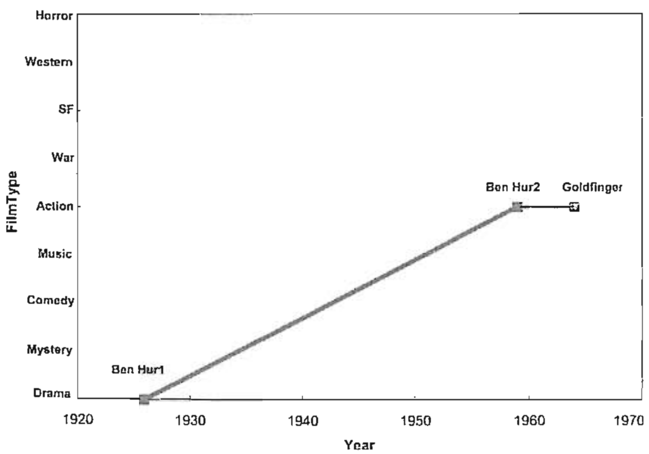


FIGURE 26.13. Mapping from data to visual form that violates expressiveness criterion.

and how they are mapped onto the Visual Structure. Note that the nominal data of the PG ratings is mapped onto a nominal visualization technique (colors). Note also, that names of directors and stars (nominal variables) are raised to ordinal variables (through alphabetization), and then mapped onto an ordinal axis. This is, of course, a common way to handle searching among a large number of nominal items.

Some properties are more effective than others for encoding information. Position is by far the most effective all-around representation. Many properties are more effective for some types of data than for others. Table 26.7 gives an approximate evaluation for the relative effectiveness of some encoding techniques based on (MacEachren, 1995). We note that spatial position is effective for all scale types of data. Shape, on the other hand, is only effective for nominal data. Gray scale is most effective for ordinal data. Such a chart can suggest representations to a visualization designer.

Taxonomy of Information Visualizations

We have shown that the properties of data and visual representation generally constrain the set of mappings that form the basis for information visualizations. Taken together, these constraints form the basis of a taxonomy of information visualizations. Such a taxonomy is given in Table 26.8. Visualizations are grouped into four categories. First are *Simple Visual Structures*, the static mapping of data onto multiple spatial dimensions, trees, or networks plus retinal variables, depicted in Fig. 26.10. Here it is worth distinguishing two cases. There is a perceptual barrier at three (or, in special cases, four) variables, a limit of the amount of data that can be perceived as an immediate whole. Bertin (1977, 1981) called this elementary unit of visual data perception the "image". Although this limit has not been definitively established in information visualization by empirical research, there must be a limit somewhere or else people could simultaneously comprehend a thousand variables. We therefore divide visualizations into those that can be comprehended in an elementary perceptual grasp (three, or in special cases, four variables)—let us call these *direct reading visualizations*—and those more complex than that barrier—which we call *articulated reading visualizations*, in which multiple actions are required.

Beyond the perceptual barrier, direct composition of data relationships in terms of 1, 2, or 3 spatial dimensions plus remaining retinal variables is still possible, but rapidly diminishes in effectiveness. In fact, the main problem of information visualization as a discipline can be seen as devising techniques for accelerating the comprehension of these more complex n-variable data relations. Several classes of techniques for n-variable visualization, which we call *Composed Visual Structures*, are based on composing Simple Visual Structures together by reusing their spatial axes. A third class of Visual Structures—*Interactive Visual Structures*—comes from using the rapid interaction capabilities of the computer. These visualizations invoke the parameter-controlling arrows of Fig. 26.10. Finally, a fourth class of visualizations—*Attention-Reactive Visual Structures*—comes from interactive displays where the system reacts to user actions

TABLE 26.6. Meta-Data and Mappings of Data onto Visual Structure in the FilmFinder

Data							*Visual Form*				
Variable	Type	Range	$Case_i$	$Case_j$	$Case_k$	...		Type	Visual Structure	Control	Transformation Affected
FilmID	N	All-IDs	230	105	540	...	→	N	Points	Button	All (details)
Title	N	All-titles	Goldfinger	Ben Hur	Ben Hur	...	→sort	O		Alphaslider	Select cases
Director	N	All-directors	Hamilton	Wyler	Niblo	...	→sort	O		Alphaslider	Select cases
Actor	N	All-actors	Connery	Heston	Novarro	...	→sort	O		Alphaslider	Select cases
Actress	N	All-actresses	Blackman	Harareet	McAvoy	...	→sort	O		Alphaslider	Select cases
Year	Q	[1926, 1989]	1964	1959	1926	...	→	Q	X-axis	Axis	Clip range
Length	Q	[0, 450]	112	212	133		→	Q		Two-sided slider	Clip range
Popularity	Q	[1, 9]	7.7	8.2	7.4	...	→	Q	Y-axis	Axis	Clip range
Rating	O	{G, PG, PG-13, R}	PG	G	G	...	→	O		Radio buttons	Select cases
Film Type	N	{Drama, Mystery, Comedy, Music, Action, War, SF, Western, Horror}	Action	Action	Drama		→	N	Color	Radio buttons	Select cases

Source: (Card et al., 1999).

TABLE 26.7. Relative Effectiveness of Position and Retinal Encodings

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

Source: (Card et al., 1999).

by changing the display, even anticipating new displays, to lower to cost of information access and sensemaking to the user. To summarize,

I. Simple Visual Structures
Direct Reading
Articulated Reading
II. Composed Visual Structures
Single-Axis Composition
Double-Axis Composition
Recursive Composition
III. Interactive Visual Structure
IV. Attention-Reactive Visual Structure

These classes of techniques may be combined to produce visualizations that are more complex. To help us keep track of the variable mapping into visual structure, we will use a simple shorthand notation for listing the element of the Visual Structure that the Data Table has mapped into. We will write, for example, [XYR^2] to note that variables map onto the X-axis, the Y-axis, and two retinal encodings. [OX] will indicate that the variables map onto one spatial axis used to arrange the objects (that is, the cases), while another was used to encode the objects' values. Examples of this notation appear in Table 26.8 and Fig. 26.21.

SIMPLE VISUAL STRUCTURES

The design of information visualizations begins with mappings from variables of the Data Table into the Visual Structure. The basic strategy for the visualization designer could be described as follows:

1. *Determine which variables of the Analytic Abstraction to map into spatial position in the Visual Structure.*
2. *Combine these mappings to increase dimensionality (e.g., by folding).*
3. *Use retinal variables as an overlay to add more dimensions.*
4. *Add controls for interaction.*
5. *Consider attention-reactive features to expand space and manage attention.*

We start by considering some of the ways in which variables can be mapped into space.

TABLE 26.8. Taxonomy of Information Visualization Techniques

I. Simple Visual Structures **Direct Reading** **1-Variable [X]** Lists 1D object charts 1D scatterplots Pie charts Folded Dimensions Distributions Box Plots **2-Variable [XY]** 2D object charts 2D scatterplots **3-Variable** [XYR] Retinal scatterplot Kahonen diagrams Retinal topographies [(XY)Z] Information landscapes Information surfaces [XYZ] 3D scatterplots **4-Variable** [XYZR] 3D retinal scatterplots 3D topographies *—Barrier of Perception—* **Articulated Reading** *n*-Variable [XYR^{n-2}] 2D Retinal scatterplots [$XYZR^{n-3}$] 2D Retinal scatterplots	Trees Node and link trees Enclosure trees TreeMaps Cone trees Networks Time ***II. Composed Visual Structures*** **Singles-Axis Composition [XY^n]** Permutation matrices Parallel coordinates **Double-Axis Composition [XY]** Graphs **Recursive Composition** 2D in 2D [$(XY)^{XY}$] Scatterplot matrices Prosection matrices Hierarchical axes Marks in 2D [$(XY)^{R}$] Stick figures Color icons Shape coding Keim spirals 3D in 3D [$(XYZ)^{XYZ}$] Worlds within worlds	***III. Interactive Visual Structures*** Dynamic queries Magic lens Overview+detail Linking and brushing Extraction & comparison Attribute xxplorer ***IV. Focus+Context Attention-Reactive Visual Abstraction*** **Data-based Methods** Filtering Selective aggregation **View-based methods** Micro-macro readings Highlighting Visual transfer functions Perspective distortion Alternate geometries

1-Variable

One-variable visual displays may actually use more than one visual dimension. This is because the data variable or attribute is displayed against some set of objects using some mark and because the mark itself takes space. Or, more subtly, it may be because one of the dimensions is used for arranging the objects and another for encoding via position the variable. A simple example would be when the data are just visually mapped into a simple text list as in Fig. 26.14(a). The objects form a sequence on the Y-dimension, and the width of the marks (the text descriptor) takes space in the X-dimension. By contrast, a one-dimensional scattergraph (Fig. 26.14[b]) does not use a dimension for the objects. Here, the Y-axis is used to display the attribute variable (suppose these are distances from home of gas stations); the objects are encoded in the mark (which takes a little bit of the X-dimension).

More generally, many single-variable visualizations are in the form $v = f(o)$, where v is a variable attribute and o is the object. Figure 26.14(c) is of this form and uses the Y-axis to encode the variable and the X-axis for the objects. Note that if the objects are, as usual, nominal, then they are reorderable: sorting the objects on the variable produces easily perceivable visual patterns. For convenience, we have used rectangular coordinates, but any other orthogonal coordinates could be used as the basis of decomposing space. Figure 26.14(d) uses θ from polar coordinates to encode, say, percentage voting for different presidential candidates. In Fig. 26.14(e), a transformation on the data side has transformed variable o into a variable representing the distribution, then mapped that onto points on the Y-axis. In Fig. 26.14(f), another transformation on the data side has mapped this distribution into 2nd quartiles, 3rd quartiles, and outlier points, which is then mapped on the visual side into a box plot on the Y-axis. Simple as they are, these techniques can be very useful, especially in combination with other techniques.

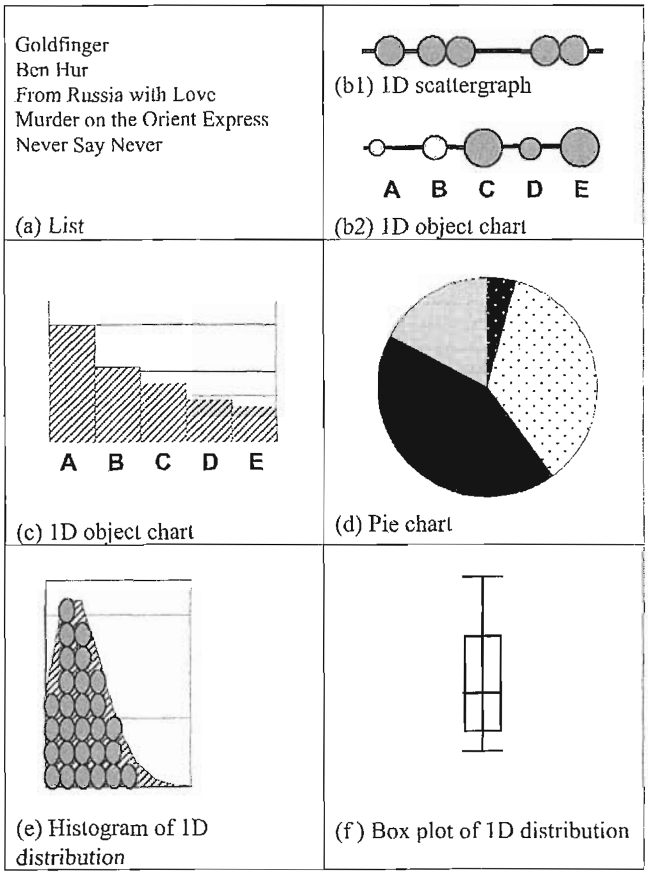


FIGURE 26.14. 1-variable visual abstractions.

One special, but common, problem is how to visualize very large dimensions. This problem occurs for single-variable visualizations, but may also occur for one dimension of a multi-variable visualization. Figure 26.15 shows several techniques for handing the problem. In Fig. 26.15(a) (Freeman & Fertig, 1995), the visual dimension is laid out in perspective. Even though each object may take only one or a few pixels on the axis, the objects are actually fairly large and selectable in the diagram. In Fig. 26.15(b) (Eick, Steffen, & Sumner, 1992), the objects (representing lines of code) are laid out on a *folded* Y-axis. When the Y-axis reaches the bottom of the page, it continues offset at the top. In Fig. 26.15(c) (Keim & Kriegel, 1994), the axis is wrapped in a square spiral. Each object is a single pixel, and its value is coded as the retinal variable color hue. The objects have been sorted on another variable; hence, the rings show the correlation of this attribute with that of the sorting attribute.

One-variable visualizations are also good parts of controls. Controls, in the form of slides, also consume considerable space on the display (for example, the controls in Fig. 26.1) that could be used for additional information communication. Figure 26.15(d) shows a slider on whose surface is a distribution representation of the number of objects for each value of the input variable, thereby communicating information about the slider's sensitivity in different data ranges. The slider on the left of Fig. 26.15(b) has a one-variable visualization that serves as a legend for the main visualization: it associates color hues with dates and allows the selection of date ranges.

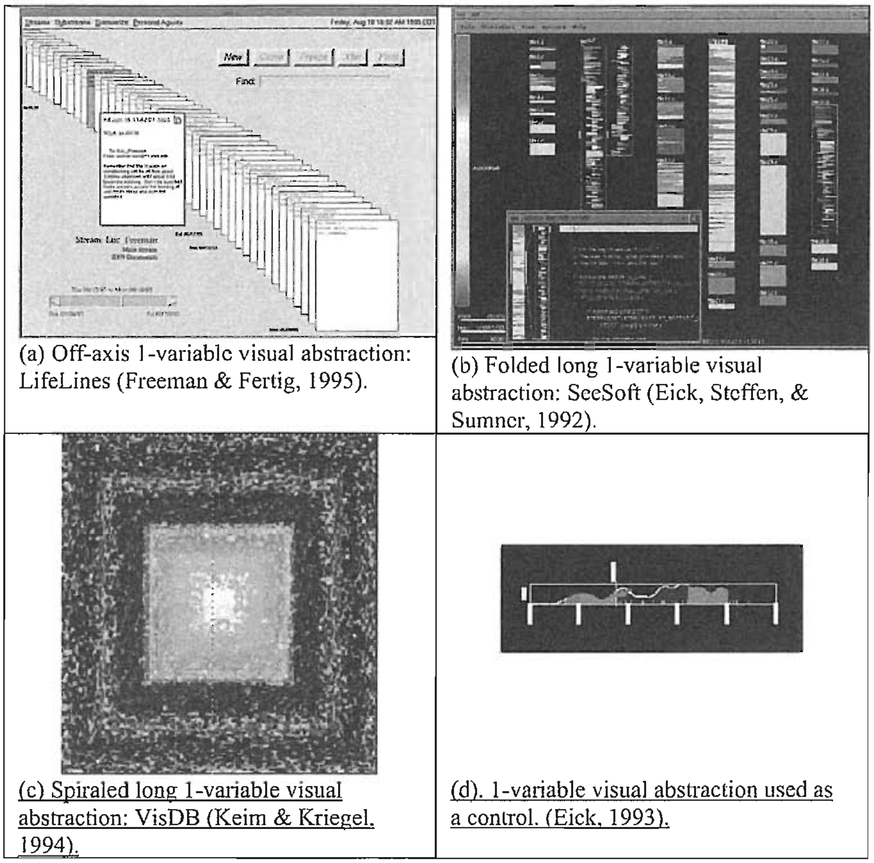
2-Variables

As we increase the number of variables, it is apparent that their mappings form a combinatorial design space. Figure 26.16 schematically plots the structure of this space, leaving out the use of multiple lower variable diagrams to plot higher variable combinations. Two-variable visualizations can be thought of as a composition of two elementary axes (Bertin, 1977, 1981; Mackinlay, 1986b), which use a single mark to encode the position on both those axes. Mackinlay called this *mark composition*, and it results in a 2D scattergraph (Fig. 26.16[g]). Note that instead of mapping onto two positional visual encodings, one positional axis could be used for the objects, and the data variables could be mapped onto a position encoding and a retinal encoding (size), as in Fig. 26.16(f).

3-Variables and Information Landscapes

By the time we get to three data variables, a visualization ca be produced in several ways. We can use three separate visual dimensions to encode the three data variables in a *3D scattergraph* (Fig. 26.16[j]). We could also use two spatial dimensions and one retinal variable in a *2D retinal scattergraph* (Fig. 26.16[k]). Or we could use one spatial dimension as an object dimension, one as a data attribute dimension, and one two retinal encodings for the other variables, as in an *object chart such as in* Fig 26.16(i). Because Fig. 26.16(i) uses multiple retinal encodings, however, it may not be as effective as other techniques. Notice that because they all encode three data variables, we have classified 2D and 3D displays together. In fact, one popular 3-variable information visualization that lies between 2D and 3D is the *information landscape* (Fig. 26.16[m]). This is essentially a 2D scattergraph with one data variable extruded into the third spatial dimension. Its essence is that two of the spatial dimensions are more tightly coupled and often relate to a 2D visualization. For example, the two dimensions might form a map with the bars showing the GDP of each region.

Another special type of 3-variable information visualization is a 2D *information topography*. In an information typography, space is partly defined by reference to external structure. For example, the topography of Fig. 26.17(a) is a map of San Francisco, requiring two spatial variables. The size of blue dots indexes the number of domain names registered to San Francisco street addresses. Looking at the patterns in the visualization shows that Internet addresses have especially concentrated in the Mission and South of Mission districts. Figure 26.17(a) uses a topography derived from real geographical space. Various techniques, such as multidimensional scaling, factor analysis, or connectionist self-organizing algorithms, can create abstract spaces based on the similarities among collections of documents or other objects. These abstract similarity spaces can function like a topography. An example can be seen in Fig. 26.17(b), where the pages in a website are depicted as regions in a similarity



(a) Off-axis 1-variable visual abstraction: LifeLines (Freeman & Fertig, 1995).

(b) Folded long 1-variable visual abstraction: SeeSoft (Eick, Steffen, & Sumner, 1992).

(c) Spiraled long 1-variable visual abstraction: VisDB (Keim & Kriegel, 1994).

(d). 1-variable visual abstraction used as a control. (Eick, 1993).

FIGURE 26.15. Uses of 1-variable visual abstractions.

space. To create this diagram[2], a web crawler crawls the site and indexes all the words and pages on the site. Each page is then turned into a document vector to represent the semantic content of that page. The regions are created using a neural network learning algorithm (see Lin, Soergel, & Marchionini (1991)). This algorithm organizes the set of web pages into regions. A visualization algorithm then draws boundaries around the regions, colors them, and names them. The result, called a *Kahonen diagram* after its original inventor, is a type of *retinal similarity topography*.

Information landscapes can also use marks that are surfaces. In Fig. 26.18(a), topics are clustered on a similarity surface, and the strength of each topic is indicated by a 3D contour. A more extreme case is Fig. 26.18(b), where an information landscape is established in spherical coordinates, and the amount of ozone is plotted as a semitransparent overlay on the ρ-axis.

n-Variables

Beyond three variables, direct extensions of the methods we have discussed become less effective. It is possible, of course to make plots using two spatial variables and n–2 retinal variables, and the possibilities for four variables are shown in Fig. 26.16. These diagrams can be understood, but at the cost of progressively more effort as the number of variables increases. It would be very difficult to understand an [XYR^{20}] retinal scattergraph, for example.

Trees

An interesting alternative to showing variable values by spatial positioning is to use explicitly drawn linkages of some kind. Trees are the simplest form of these. Trees map cases into subcases. One of the data variables in a Data Table (for example, the variable ReportsTo in an organization chart) is used to define the tree. There are two basic methods for visualizing a tree: (a) Connection and (b) Enclosures.

Connection. Connection uses lines to connect marks signifying the nodes of the tree. Logically, a tree could be drawn merely by drawing lines between objects located randomly positioned on the plane, but such a tree would be visually unreadable. Positioning in space is important. Figure 26.20(a) is a tree from Charles Darwin's notebook (Robin, 1992) drawn to help

[2]This figure is produced by a program called SiteMap by Xia Lin and associates. See http://faculty.cis.drexel.edu/sitemap/index.html.

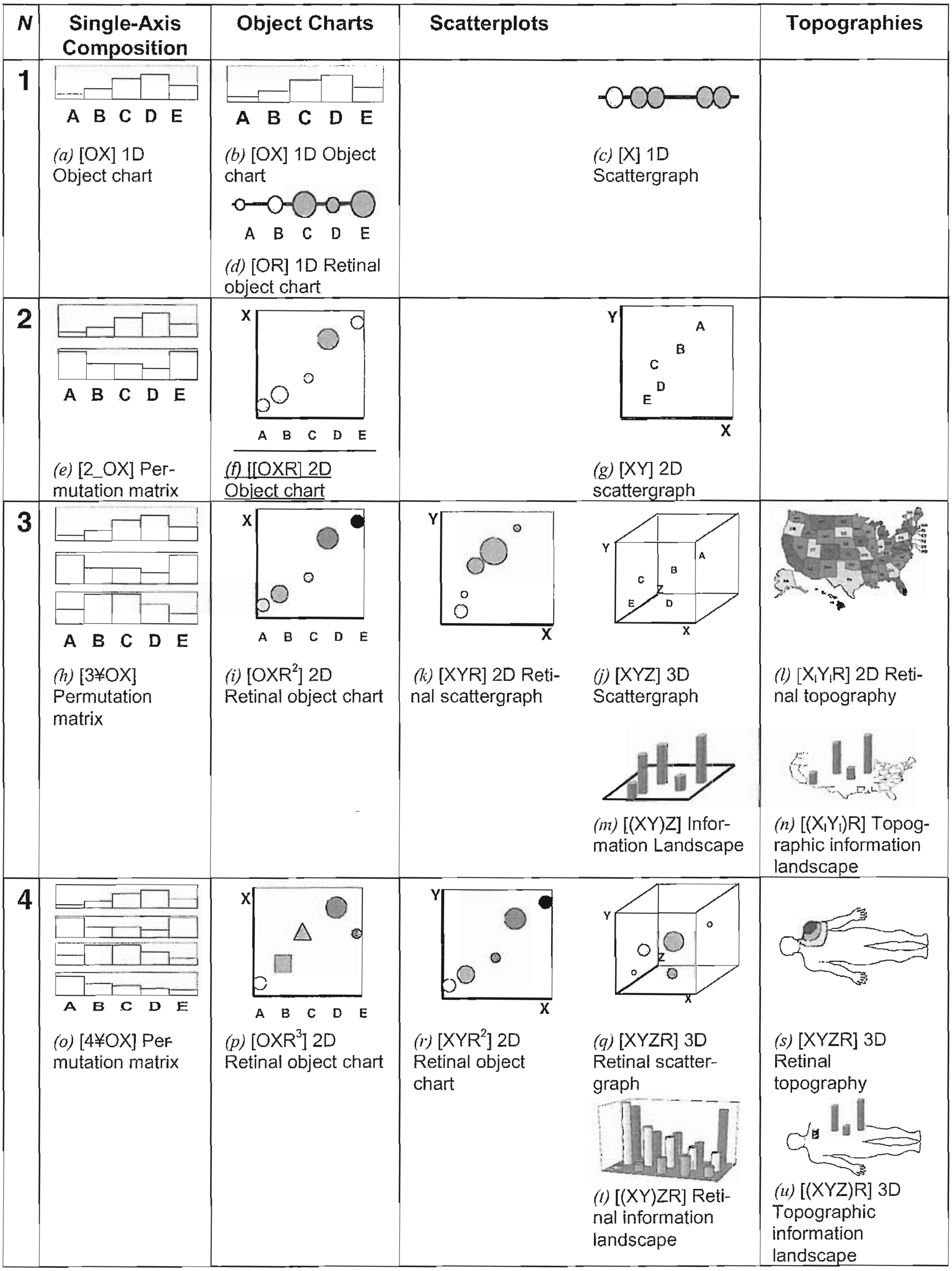
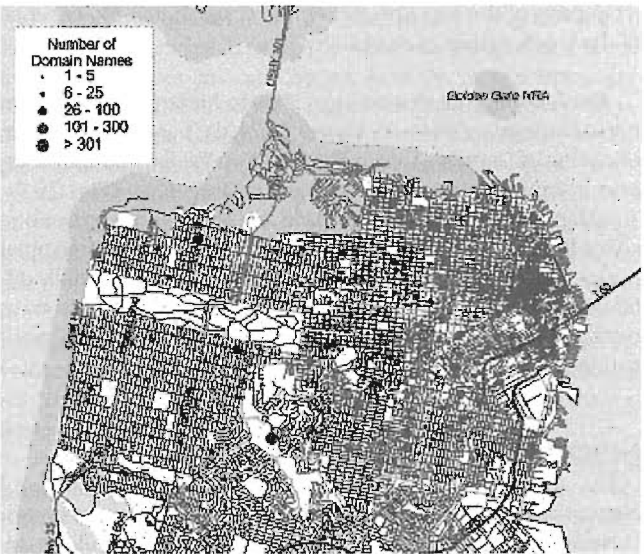
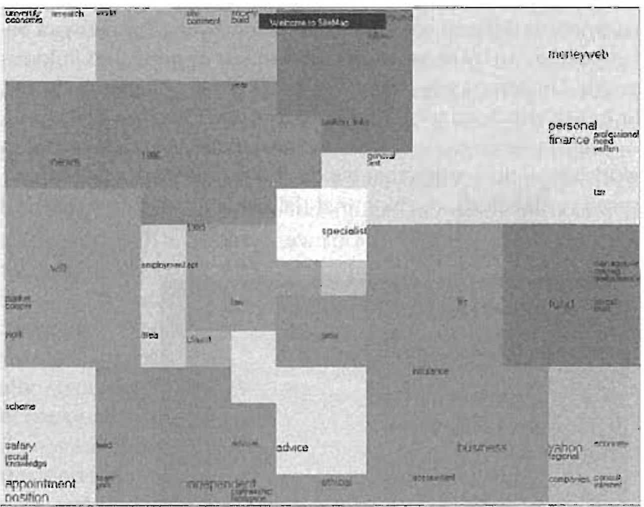


FIGURE 26.16. Simple Visual Structures..



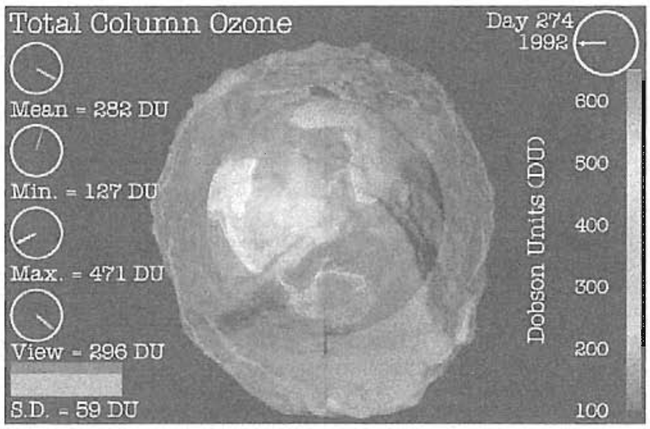
(a) $X_I Y_I R$ Retinal topography



(b) $X_S Y_S R$ Retinal similarity topography

FIGURE 26.17. Retinal information topographies.

(a) News stories based on ThemeScapes (Wise et al., 1995). Courtesy NewsMaps.com.

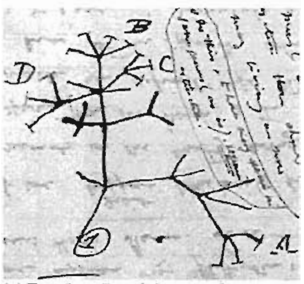


(b) Ozone layer surrounding earth. L. Treinish. Courtesy IBM.

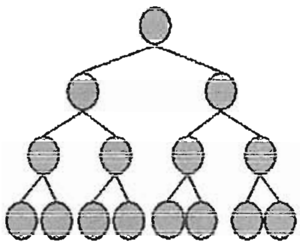
FIGURE 26.18. 3D information surface topographies.

him work out the theory of evolution. Lines proceed from ancestor species to new species. Note that even in this informal setting intended for personal use that the tree uses space systematically (and opportunistically). There are no crossed lines. A common way of laying out trees is to have the depth in the tree map onto one ordinal access as in Fig. 26.20(b), while the other axis is nominal and used to separate nodes. Of course, trees could also be mapped into other coordinate systems: for example, there can be circular trees in which the r-axis represents depth and the θ-axis is used to separate nodes as in the representation of the evolution species in Fig. 26.20(c).[3] It is because trees have no cycles that one of the spatial dimensions can be used to encode tree depth. This partial correlation of tree structure and space makes trees relatively easy to lay out and interpret, compared to generalized networks. Hierarchical displays are important not only because many interesting collections of information, such as organization charts or taxonomies, are hierarchical data, but also because important collections of information, such as websites, are *approximately* hierarchical. Whereas practical methods exist for displaying trees up to several thousand nodes, no good methods exist for displaying general graphs of this size. If a visualization problem involves the displaying of network data, a practical design heuristic is to see whether the data might not be forced into a display as a modified tree, such as a tree with a few non-tree links. A significant disadvantage of trees is that as they get large, they acquire an extreme aspect ratio, because the nodes expand exponentially as a function with depth. Consequently, any sufficiently large tree (say, >1000 nodes) resembles a straight line. Circular trees such as Fig. 26.20(c) are one way of trying to buy more space to mitigate this problem. Another disadvantage of trees is the significant empty space between nodes to make their organization easily readable. Various tricks can be used to

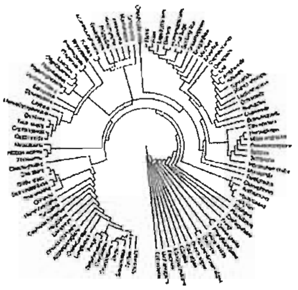
[3] This figure is from David Hillis, University of Texas.



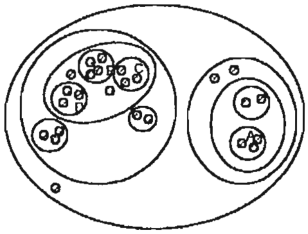
(a) Tree from Darwin's notes, from (Robin, 1992). Courtesy Syndics of Cambridge University Library, Cambridge, UK.



(b) Typical link and node tree layout.



(c) Circular tree of evolution of life. Courtesy David Hillis, University of Texas.



(d) Tree in (a) drawn using enclosure.

FIGURE 26.19. Trees.

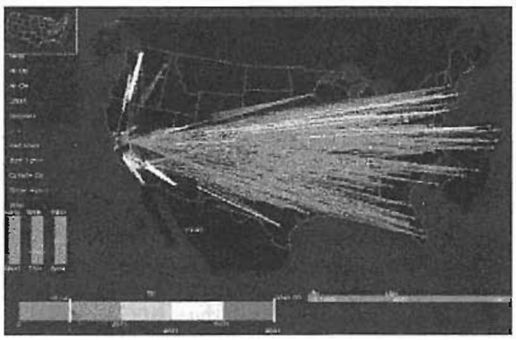
wrap parts of the tree into this empty space, but at the expense of the tree's virtues of readability.

Enclosure. Enclosure uses lines to hierarchically enclose nested subsets of the tree. Figure 26.20(d) is an enclosure tree encoding of Darwin's tree in Fig. 26.20(a). We have already seen one attempt to use tree enclosure, TreeMaps (Fig. 26.5). TreeMaps make use of all the space and stays within prescribed space boundaries, but they do not represent the nonterminal nodes of the tree very well and similar leaves can have wildly different aspect ratios. Recent variations on TreeMaps found ways to "squarify" nodes (Shneiderman & Wattenberg, 2001), mitigating this problem.

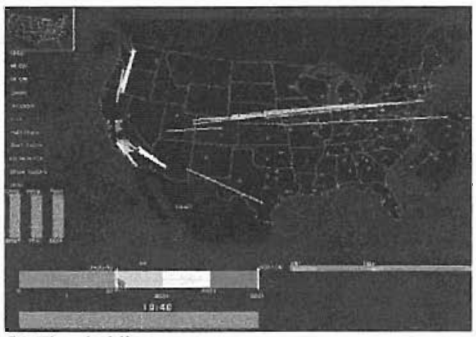
Networks

Networks are more general than trees and may contain cycles. Networks may have directional links. They are useful for describing communication relationships among people, traffic in a telephone network, and the organization of the Internet. Containment is difficult to use as a visual encoding for network relationships, so most networks are laid out as node and link diagrams. Unfortunately, straightforward layouts of large node and link diagrams tend to resemble a large wad of tangled string.

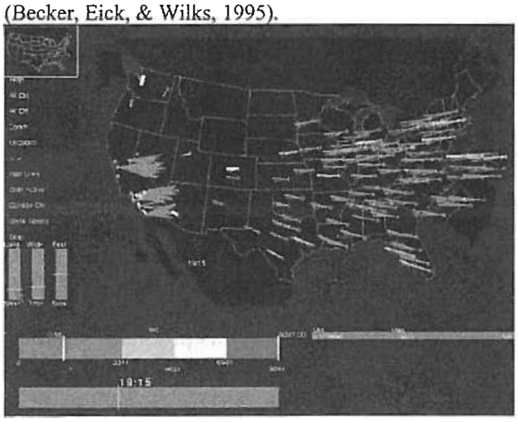
We can distinguish the same types of nodes and links in network Visual Structures that we did for spatial axes: (a) Unstructured (unlabeled), (b) Nominal (labeled), (c) Ordinal (labeled



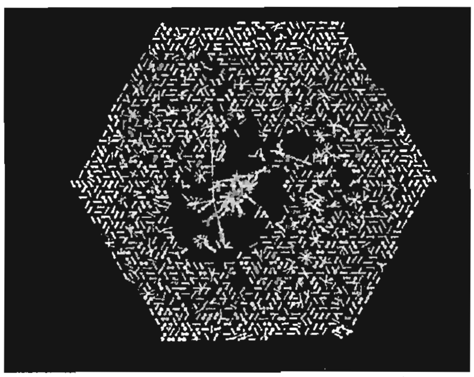
(a) Telephone traffic after California earthquake. (Becker, Eick, & Wilks, 1995).



(b) Thresholding.



(c) Line shortening



(d) Visualization to detect telephone fraud. (Cox, Eick, & Wills, 1997).

FIGURE 26.20. Network methods.

with an ordinal quantity), or (d) Quantitative (weighted links). Retinal properties, such as size or color, can be used to encode information about links and nodes. As in the case of trees, spatial positioning of the nodes is extremely important. Network visualizations escape from the strong spatial constraints of simple Visual Structures only to encounter another set of strong spatial constraints of node links crossing and routing. Networks and trees are not so much an alternative of direct of the direct graphical mappings we have discussed so far as they are another set of techniques that can be overlaid on these mappings. Small node and link diagrams can be laid out opportunistically by hand or by using graph drawing algorithms that have been developed (Battista, Eades, Tamassia, & Tollis, 1994; Cruz & Tamassia, 1998; Tamassia, 1996) to optimize minimal link crossing, symmetry, and other aesthetic principles.

For very large node and link diagrams, additional organizing principles are needed. If there is an external topographic structure, it is sometimes possible to use the spatial variables associated with the nodes. Figure 26.20(a) shows a network based on call traffic between cities in the United States (Becker, Eick, & Wilks, 1995). The geographical location of the cities is used to lay out the nodes of the network. Another way to position nodes is by associating nodes with positions in a similarity space, such the nodes that have the strongest linkages to each other are closest together. There are several methods for computing node nearness in this way. One is to use multidimensional scaling (MDS) (Fairchild, Poltrock, & Furnas, 1988). Another is to use a "spring" technique, in which each link is associated with a Hooke's Law spring weighted by strength of association and the system of springs is solved to obtain node position. Eick and Willis (1993) have argued that the MDS technique places too much emphasis on smaller links. They have derived an alternative that gives clumpier (and hence, more visually structured) clusters of nodes. If positioning of nodes corresponds perfectly with linkage information, then the links do not add more visual information. If positioning does not correspond at all with linkage information, then the diagram is random and obscure. In large graphs, node positions must have a partially correlated relationship to linkage in order to allow the emergence of visual structure. Note that this is what happens in the telephone traffic diagram Fig. 26.20(a). Cities are positioned by geographical location. Communication might be expected to be higher among closer cities, so the fact that communications is heavy between coasts stands out.

A major problem in a network such as Fig. 26.20(a) is that links may obscure the structure of the graph. One solution is to route the links so that they do not obscure each other. The links could even be drawn outside the plane in the third dimension; however, there are limits to the effectiveness of this technique. Another solution is to use *thresholding*, as in Fig. 26.20(b). Only those links representing traffic greater than a certain threshold are included; the others are elided allowing us to see the most important structure. Another technique is *line shortening*, as in Fig. 26.20(c). Only the portion of the line near the nodes is drawn. At the cost of giving up the precise linkage, it is possible to read the density of linkages for the different nodes. Figure 26.20(d) is a technique used to find patterns in an extremely large network. Telephone subscribers are represented as nodes on a hexagonal array. Frequent pairs are located near each other on the array. Suspicious patterns are visible because of the sparseness of the network.

The insightful display of large networks is difficult enough that many information visualization techniques depend on interactivity. One important technique, for example, is node aggregation. Nodes can be aggregated to reduce the number of links that have to be drawn on the screen. Which nodes are aggregated can depend on the portion of the network on which the user is drilling down. Similarly, the sets of nodes can be interactively restricted (e.g., telephone calls greater than a certain volume) to reduce the visualization problem to one within the capability of current techniques.

COMPOSED VISUAL STRUCTURES

So far, we have discussed simple mappings from data into spatial position axes, connections and enclosures, and retinal variables. These methods begin to run into a barrier around three variables as the spatial dimensions are used up and as multiple of the less efficient retinal variables needed. Most interesting problems involve many variables. We shall therefore look at a class of methods that reuse precious spatial axes to encode variables. This is done by composing a compound Visual Structure out of several simple Visual Structures. We will consider five subclasses of such composition: (a) mark composition, (b) case composition, (c) single-axis composition, (d) double-axis composition, and (e) recursive composition. Schematically, we illustrate these possibilities in Fig. 26.21.

Single-axis composition. In single-axis composition, multiple variables that share a single axis are aligned using that axis, as illustrated in Fig. 26.21(a). An example of single-axis composition is a method due to Bertin called *permutation matrices* (Bertin, 1977/1981). In a permutation matrix (Fig. 26.16[o], for example), one of the spatial axes is used to represent the cases and the other a series of bar charts (or rows of circles of different size or some other depiction of the value of each variable) to represent the values. In addition, bars for values below average may be given a different color, as in Fig. 26.7, in order to enhance the visual patterns. The order of the objects and the order of the variables may both be permuted until patterns come into play. Permutation matrices were used in our hotel analysis example. They give up direct reading of the data space in order to handle a larger number of variables. Of course, as the number of variables (or objects) increases, manipulation of the matrices becomes more time-consuming and visual interpretation more complex. Still, permutation matrices or their variants are one of the most practical ways of representing multi-variable data.

If we superimpose the bar charts of the permutation matrix atop one another, and then replace the bar chart with a line linking together the tops of the bars, we get another method for handling multiple variables by single-axis composition—*parallel coordinates* (Inselberg, 1997; Inselberg & Dimsdale, 1990), as shown in Fig. 26.22. A problem is analyzed in parallel coordinates by interactively restricting the objects displayed (the lines) in order to look at cases with common characteristics. In Fig.

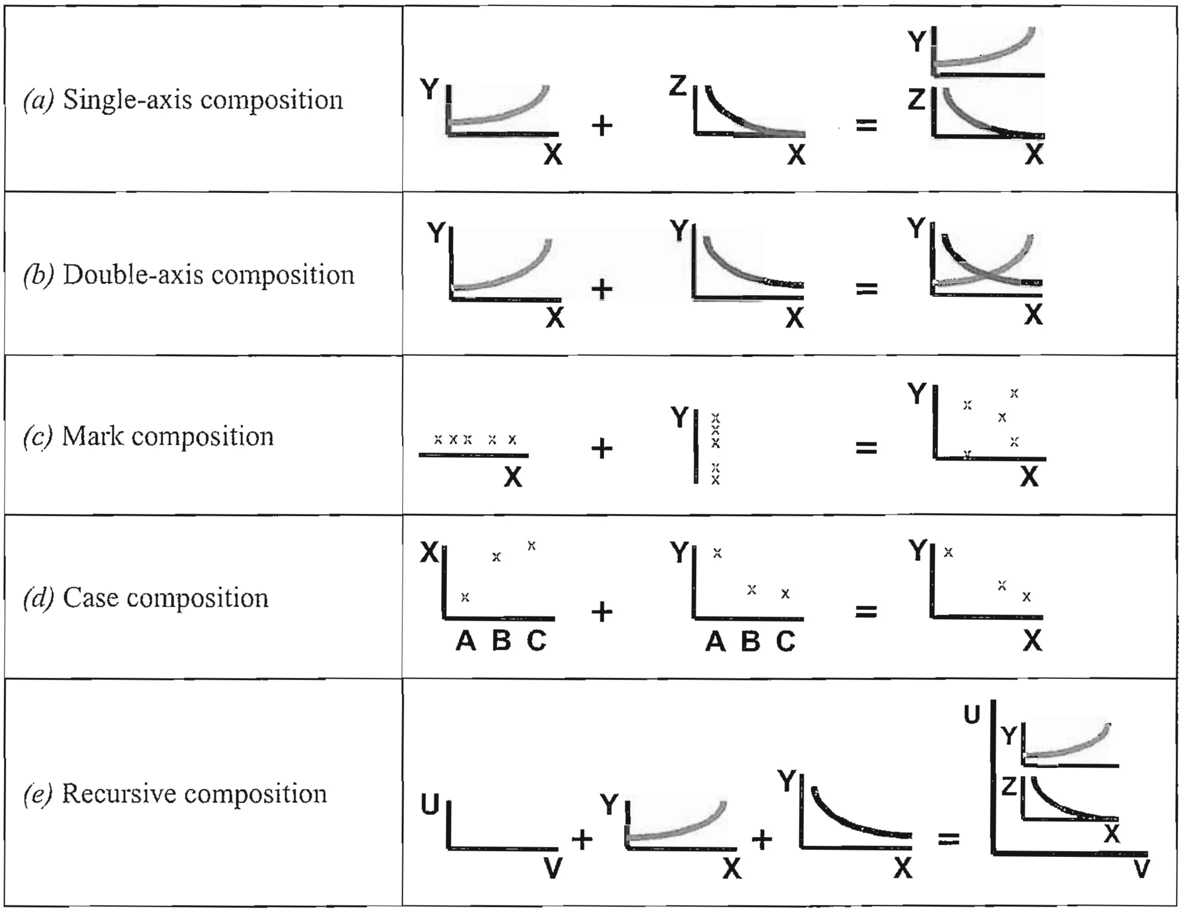


FIGURE 26.21. Composition types.

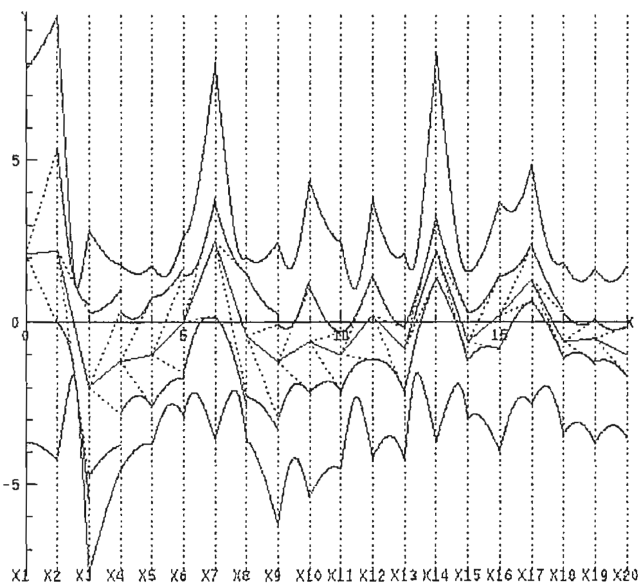


FIGURE 26.22. Single-axis composition: parallel coordinates.

26.22 , parallel coordinates are used to analyze the problem of yield from a certain processor chip. X1 is chip yield, X2 is quality, X3 through X12 are defects, and the rest of the variables are physical parameters. The analysis, looking at those subsets of data with high yield and noticing the distribution of lines on the other parameters, was able to solve a significant problem in chip processing.

Both permutation matrices and parallel coordinates allow analyses in multi-dimensional space, because they are efficient in the use (and reuse) of spatial position and the plane. Actually, they also derive part of their power from being interactive. In the case of permutation matrices, interactivity comes in re-ordering the matrices. In the case of parallel coordinates, interactivity comes in selecting subsets of cases to display.

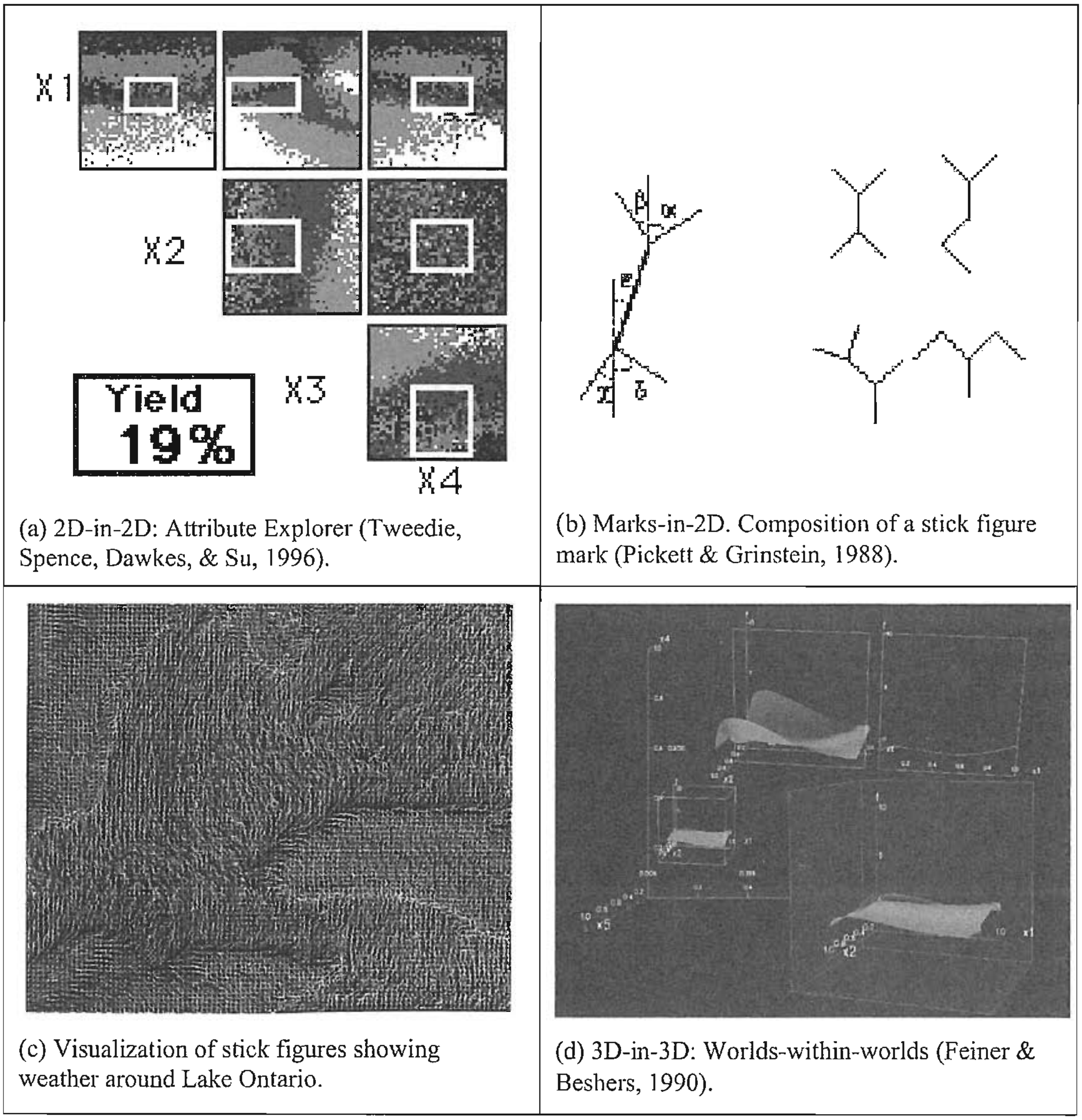
Double-axis composition. In double-axis composition, two visual axes must be in correspondence, in which case the cases are plotted on the same axes as a multivariable graph (Fig. 26.21[b]). Care must be taken that the variables are plotted on a comparable scale. For this reason, the separate scales of the variables are often transformed to a common proportion change scale. An example would be change in price for various stocks.

The cases would be the years, and the variables would be the different stocks.

Mark composition and case composition. Composition can also fuse diagrams. We discussed that each dimension of visual space can be said to have properties as summarized in Table 26.4. The visual space of a diagram is composed from the properties of its axis. In *mark composition* (Fig. 26.21[c]), the mark on one axis can fuse with the corresponding mark on another axis to form a single mark in the space formed by the two axes. Similarly, two object charts can be fused into a single diagram by having a single mark for each case. We call this latter form *case composition* Fig. 26.21(d).

Recursive composition. Recursive composition divides the plane (or 3D space) into regions, placing a subvisualization in each region (Fig. 26.21[e]). We use the term somewhat loosely, since regions have different types of subvisualizations. The FilmFinder in Fig. 26.1 is a good example of a recursive visualization. The screen breaks down into a series of simple Visual Structures and controls: (a) a 3-variable retinal scattergraph (Year, Rating, FilmType) + (b) a 1-variable slider (Title) + (c) a 1-variable slider (Actors) + (d) a 1-variable slider (Actresses) + (e) a 1-variable slider (Director) + (f) a 1-variable slider (FilmLength) + (g) a 1-variable radio button control (Rating) + (h) a 1-variable button-set (FilmType).

Three types of recursive composition deserve special mention: (a)2D-in-2D, (b) marks-in-2D, and (c) 3D-in-3D. An example of *2D-in-2D composition* is the "prosection matrix" (Tweedie, Spence, Dawkes, & Su, 1996) shown in Fig. 26.23(a). Each smaller square in the prosection matrix represents a pair of parameters plotted against each other. The coloring shows



(a) 2D-in-2D: Attribute Explorer (Tweedie, Spence, Dawkes, & Su, 1996).

(b) Marks-in-2D. Composition of a stick figure mark (Pickett & Grinstein, 1988).

(c) Visualization of stick figures showing weather around Lake Ontario.

(d) 3D-in-3D: Worlds-within-worlds (Feiner & Beshers, 1990).

FIGURE 26.23. Recursive composition.

which values of the plotted pair give excellent (red region) or partly good (gray regions) performance for the design of some device. The arrangement of the individual matrices into a supermatrix redefines the spatial dimensions (that is, associates it with different variables) within each of the cells, and the cells themselves are arranged in an overall scheme that systematically uses space. In this way, the precious spatial dimension is effectively expanded to where all the variables can reuse it. An important property of techniques similar to this one is that space is defined at more than one *grain size*, and these levels of grain become the basis for a *macro-micro reading*.

An example of *marks-in-2D composition* in the use of "stick figure" displays. This is an unusual type of visualization in which the recursion is within the mark instead of within the use of space. Figure 26.23(b) shows a mark that is itself composed of submarks. The mark is a line segment with four smaller line segments protruding from the ends. Four variables are mapped onto angle of these smaller line segments and a fifth onto the angle of the main line segment. Two additional variables are mapped onto the position of this mark in a 2D display. A typical result is the visualization in Fig. 26.23(c), which shows five weather variables around Lake Ontario, the outline of which clearly appears in the figure.

Feiner and Beshers (1990) provided an example of the third recursive composition technique, *3D-in-3D composition*. Suppose a dependent variable is a function of six continuous variables, $y = f(x, y, z, w, r, s)$. Three of these variables are mapped onto a 3D coordinate system. A position is chosen in that space, say, $x1, y1, z1$. At that position, a new 3D coordinate system is presented with a surface defined by the other three variables (Fig. 26.23[d]). The user can thus view $y = f(x1, y1, z1, w, r, s)$. The user can slide the second-order coordinate system to any location in the first, causing the surface to change appropriately. Note that this technique combines a composed visual interaction with interactivity on the composition. Multiple second-order coordinate systems can be displayed at the space simultaneously, as long as they do not overlap by much.

INTERACTIVE VISUAL STRUCTURES

In the examples we have considered so far, we have often seen that information visualization techniques were enhanced by being interactive. Interactivity is what makes visualization a new medium, separating it from generations of excellent work on scientific diagrams and data graphics. Interactivity means controlling the parameters in the visualization reference model (Fig. 26.10). This naturally means that there are different types of interactivity, because the user could control the parameters to data transformations, to visual mappings, or to view transformations. It also means that there are different forms of interactivity based on the response cycle of the interaction. As an approximation, we can think of there being three time constants that govern interactivity, which we take to be 0.1 sec, 1 sec, and 10 sec (Card, Moran, & Newell, 1986) (although the ideal value of these may be somewhat less, say, 0.07 sec, 0.7 sec, and 7 sec). The first time constant is the time in which a system response must be made, if the user is to feel that there is a direct physical manipulation of the visualization. If the user clicks on a button or moves a slider, the system needs to update the display in less than 0.1 sec. Animation frames need to take less than 0.1 sec. The second time constant, 1 sec, is the time to complete an immediate action, for example, an animated sequence such as zooming in to the data or rotating a tree branch. The third time constant 10 sec (meaning somewhere in the 5 to 30 sec interval) is the time for completing some cognitive action, for example deleting an element from the display. Let us consider a few well-known techniques for interactive information visualizations.

Dynamic queries. A general paradigm for visualization interaction is dynamic queries, the interaction technique used by the FilmFinder in Fig. 26.1. The user has a visualization of the data and a set of controls, such as sliders, by which subsets of the Data Table can be selected. For example, Table 26.9 shows the mappings of the Data Table and controls for the FilmFinder. The sliders and other controls will select which subset of the data is going to be displayed. In the FilmFinder, the control for Length is a two-sided slider. Setting one end to 90 minutes and the other end to 120 minutes will select for display only those cases of the Data Table whose year variable lies between these limits. The display needs to change within the 0.1 sec of changing the slider.

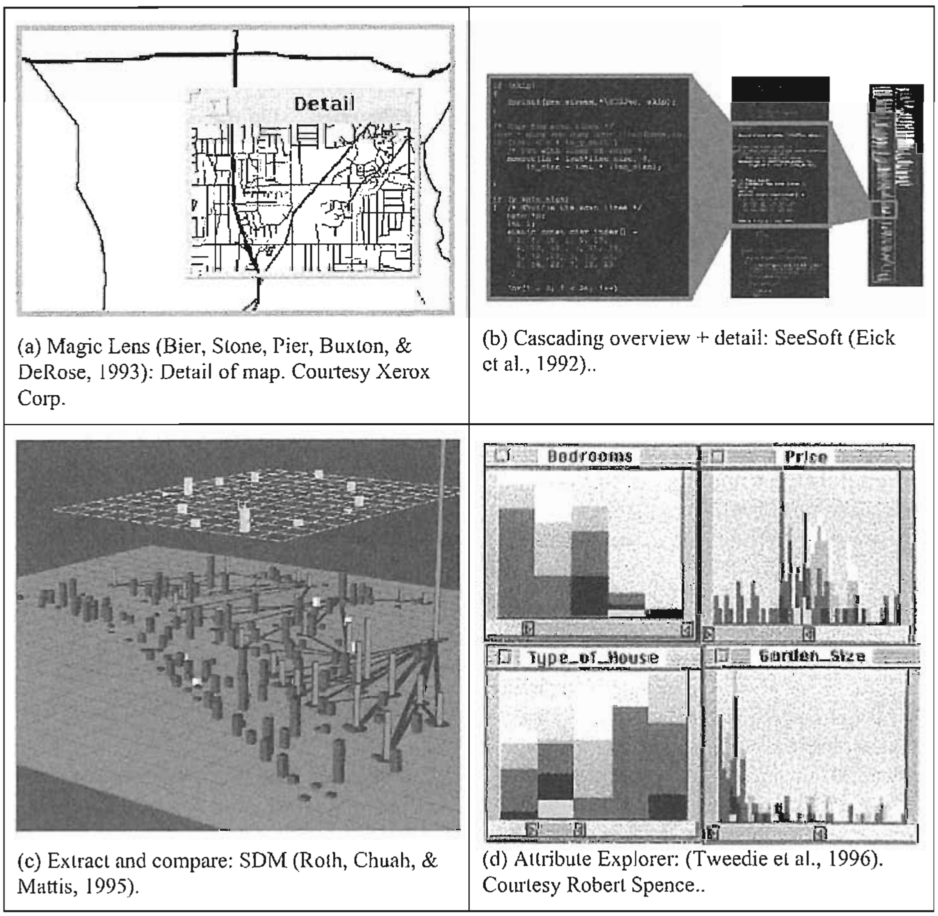
Magic lens (movable filter). Dynamic queries is one type of interactive filter. Another type is a movable filter that can be moved across the display, as in Fig. 26.24(a). These *magic lenses* are useful when it is desired to filter only some of the display. For example, a magic lens could be used with a map that showed the population of any city it was moved over. Multiple magic lenses can be used to cascade filters.

Overview + detail. We can think of an overview + detail display (Fig. 26.24[b]) as a particular type of magic lens, one that magnifies the display and has the magnified region off to the side so as not to occlude the region. Displays have information at different grain sizes. A GIS map may have information at the level of a continent as well as at the level of a city. If the shape of the continent can be seen, the display is too coarse to see the roadways of a city. Overview + detail displays show that data at more than one level, but they also show where the finer grain display fits into the larger grain display. In Fig. 26.24(b), from SeeSoft (Eick et al., 1992), a system for visualizing large software systems, the amount of magnification in the detail view is large enough that two concatenated overview + detail displays are required. Overview + detail displays are thus very helpful for data navigation. Their main disadvantage is that the require coordination of two visual domains.

Linking and brushing. Overview + detail is an example of coordinating dual representations of the same data. These can be coordinated interactively with *linking and brushing*. Suppose, for example, we wish to show power consumption on an airplane, both in terms of the physical representation of the airplane and a logical circuit diagram. The two views could be shown and *linked* by using the same color for the same component types. Interactivity itself can be used for a dynamic form of linking called *brushing*. In brushing, running the

TABLE 26.9. Visual Marks and Controls for FilmFinder

Data							Visual Form				
Variable	Type	Range	$Case_i$	$Case_j$	$Case_k$	...		Type	Visual Structure	Control	Transformation Affected
FilmID	N	All-IDs	230	105	540	...	→	N	Points	Button	All (details)
Title	N	All-titles	Goldfinger	Ben Hur	Ben Hur	...	→sort	O		Alphaslider	Select cases
Director	N	All-directors	Hamilton	Wyler	Niblo	...	→sort	O		Alphaslider	Select cases
Actor	N	All-actors	Connery	Heston	Novarro	...	→sort	O		Alphaslider	Select cases
Actress	N	All-actresses	Blackman	Harareet	McAvoy	...	→sort	O		Alphaslider	Select cases
Year	Q	[1926, 1989]	1964	1959	1926	...	→	Q	X-axis	Axis	Clip range
Length	Q	[0, 450]	112	212	133		→	Q		Two-sided slider	Clip range
Popularity	Q	[1, 9]	7.7	8.2	7.4	...	→	Q	Y-axis	Axis	Clip range
Rating	O	{G, PG, PG-13, R}	PG	G	G	...	→	O		Radio buttons	Select cases
Film Type	N	{Drama, Mystery, Comedy, Music, Action, War, SF, Western, Horror}	Action	Action	Drama		→	N	Color	Radio buttons	Select cases



(a) Magic Lens (Bier, Stone, Pier, Buxton, & DeRose, 1993): Detail of map. Courtesy Xerox Corp.

(b) Cascading overview + detail: SeeSoft (Eick et al., 1992)..

(c) Extract and compare: SDM (Roth, Chuah, & Mattis, 1995).

(d) Attribute Explorer: (Tweedie et al., 1996). Courtesy Robert Spence..

FIGURE 26.24. Interaction techniques.

cursor over a part of one of the views causes highlighting both in that view and in the other view.

Extraction and comparison. We can also use interaction to extract a subset of the data to compare with another subset. An example of this is in the SDM system (Chuah, Roth, Mattis, & Kolojejchick, 1995) in Fig. 26.24(c). The data are displayed in a 3D information landscape, but the perspective interferes with the ability to compare it. Information is therefore *extracted* from the display (leaving ghosts behind) and placed in an orthogonal viewing position where it can be *compared* using 2D. It could also be dropped into another display. Interactivity makes possible these manipulations, while keeping them coordinated with the original representations.

Attribute explorer. Several of these interactive techniques are combined in the Attribute Explorer (Tweedie et al., 1996). Figure 26.24(d) shows information on four attributes of houses. Each attribute is displayed by a histogram, where each square making up the histogram represents an individual house. The user selects a range of some attribute, say price. Those pixels making up the histogram on price have their corresponding pixels linked representing houses highlighted on the other attributes. Those houses meeting all the criteria are highlighted in one color; those houses meeting, say, all but one are highlighted in another color. In this way, the user can tell about the "near misses." If the users were to relax one of the criteria only a little (say, reducing price by $100), then the user might be able to gain more on another criterion (say, reducing a commute by 20 miles).

FOCUS + CONTEXT ATTENTION-REACTIVE ABSTRACTIONS

So far, we have considered visualizations that are static mappings from Data Table to Visual Structure and those where the mappings Data Table to Visual Structure are interactively controlled by the user. We now consider visualizations in which the machine is no longer passive, but its mappings from Visual Structure to View are altered by the computer according to the its model of the user's *degree of interest*. We can, in principle, associate a cost of access with every element in the Data Table. Take the FilmFinder in Figure 26.3. Details about the movie "Murder on the Orient Express" are accessible at low cost in terms of time because they are presently visible on the screen. Details of "Goldfinger," a movie with only a mark on the display, take more time to find. Details of "Last Year at Marienbad," a movie with no mark on the display, would take much more time. The idea is that with a model for predicting users' changes in interest, the system can adjust its displays to make costs lower for information access. For example, if the user wants some detail about a movie, such as the director, the system can anticipate that the user is more likely to want other details about the movie as well and therefore display them all at the same time: The user does not have execute a separate command; the cost is therefore reduced.

Focus+context views are based on several premises: First, the user needs both overview (context) and detail information (focus) during information access, and providing these in separate screens or separate displays is likely to cost more in user time. Second, information needed in the overview may be different from that needed in the detail. The information of the overview needs to provide enough information the user to decide where to examine next or to give a context to the detailed information rather than the detailed information itself. As Furnas (1981) has argued, the user's interest in detail seems to fall away in a systematic way with distance as information objects become farther from current interest. Third, these two types of information can be combined within a single dynamic display, much as human vision uses a two-level focus and context strategy. Information broken into multiple displays (separate legends for a graph, for example) seem to degrade performance due to reasons of visual search and working memory.

Furnas (1981) was the first to articulate these ideas systematically in his theory of *fisheye views*. The essence of focus+context displays is that the average cost of accessing information is reduced by placing the most likely needed information for navigation and detail where it is fastest to access. This can be accomplished by working on either the data side or the visual side of the visual reference model, Fig. 26.10. We now consider these techniques in more detail.

Data-Based Methods

Filtering. On the data side, focus+context effects can be achieved by filtering out which items from the Data Table are actually displayed on the screen. Suppose we have a tree of categories taken from Roget's Thesaurus, and we are interacting with one of these, "Hardness."

Matter
  ORGANIC
    Vitality
      Vitality in general
      Specific vitality
    Sensation
      Sensation in general
      Specific sensation
  INORGANIC
    Solid
      Hardness
      Softness
    Fluid
      Fluids in general
      Specific fluids

Of course, this is a small example for illustration. A tree representing a program listing or a computer directory or a taxonomy could easily have thousands of lines, a number that would vastly exceed what could fit on the display and hence would have a high cost of accessing. We calculate a degree-of-interest (DOI) for each item of the tree, given that the focus is on the node Hardness. To do this, we split the DOI into an intrinsic part and a part that varies with distance from the current center of interest and use a formula from Furnas (1981).

$$DOI = Intrinsic\ DOI + Distance\ DOI$$

Figure 26.25 shows schematically how to perform this computation for our example. We assume that the intrinsic DOI of a node is just its distance of the root (Fig. 26.25[a]). The distance part of the DOI is just the traversal distance to a node from the current focus node (Fig. 26.25[b]; it turns out to be convenient to use negative numbers for this computation, so that the maximum amount of interest is bounded, but not the minimum amount of interest). We add these two numbers together (Fig. 26.25 [c]) to get the DOI of each node in the tree. Then we apply a minimum threshold of interest (−5 in this case) and only show nodes more interesting than that threshold. The result is the reduced tree:

Matter
  INORGANIC
  ORGANIC
    Solid
      Hardness
      Softness
    Fluid

The reduced tree gives local context around the focus node and progressively less detail farther away. But it does seem to give the important context.

Selective aggregation. Another focus+context technique from the data side is selective aggregation. Selective aggregation creates new cases in the Data Table that are aggregates of other cases. For example, in a visualization of voting behavior in a presidential election, voters could be broken down by sex, precinct, income, and party affiliation. As the user drills down on, say, male Democrats earning between $25,000 and $50,000, other categories could be aggregated, providing screen space and contextual reference for the categories of immediate interest.

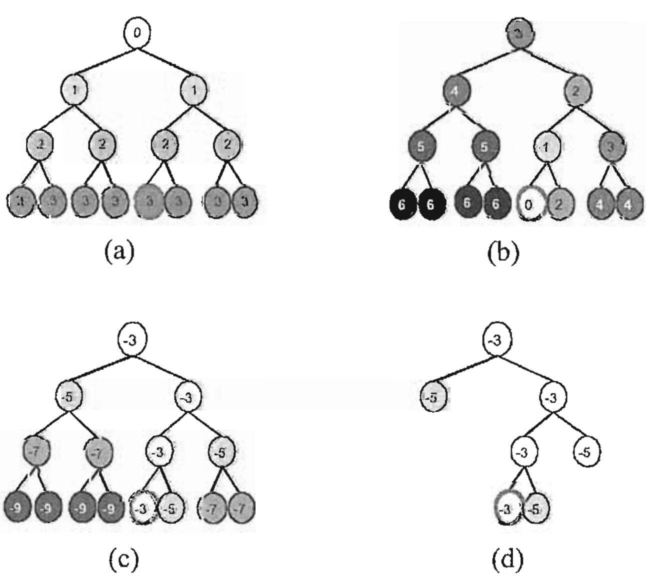


FIGURE 26.25. Degree-of-Interest calculation for fish-eye visualization.

View-Based Methods

Micro-macro readings. Micro-macro readings are diagrams in which "detail cumulates into larger coherent structures" (Tufte, 1990). The diagram can be graphically read at the level of larger contextual structure or at the detail level. An example is Fig. 26.26. The micro reading of this diagram shows three million observations of the sleep (lines), wake (spaces), and feeding (dots) activity of a newborn infant. Each day's activity is repeated three times on a line to make the cyclical aspect of the activity more clearly visible. The macro reading of the diagram, emphasized the thick lines, shows the infant transitioning from the natural human 25-hour cycle at birth to the 24-hour solar day. The macro reading serves as context and index into the micro reading.

Highlighting. Highlighting is a special form of micro-macro reading in which focal items are made visually distinctive in some way. The overall set of items provides a context for the changing focal elements.

Visual transfer functions. We can also warp the view with viewing transformations. An example is a visualization called the *bifocal lens* (Spence & Apperley, 1982). Fig. 26.27(a) shows a set of documents the user would like to view, but which is too large to fit on the screen. In a bifocal lens, documents not in a central focal region are compressed down to a smaller size. This could be a strict visual compression. It could also involve a change in representation. We can talk about the visual compression in terms of a visual transfer function Fig. 26.27(b), sometimes conveniently represented in terms of its first derivative in Fig. 26.27(c). This function shows how many units of an

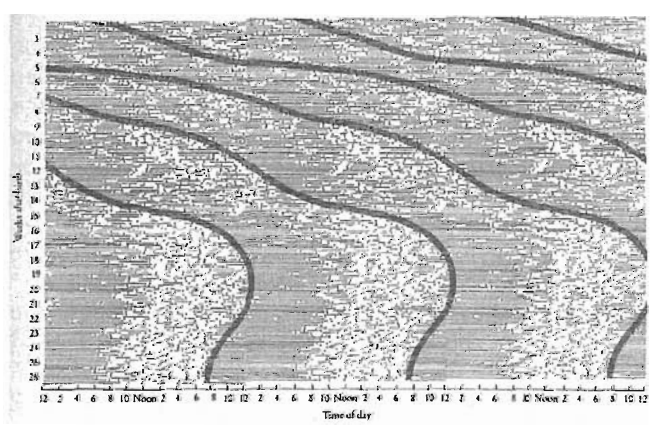
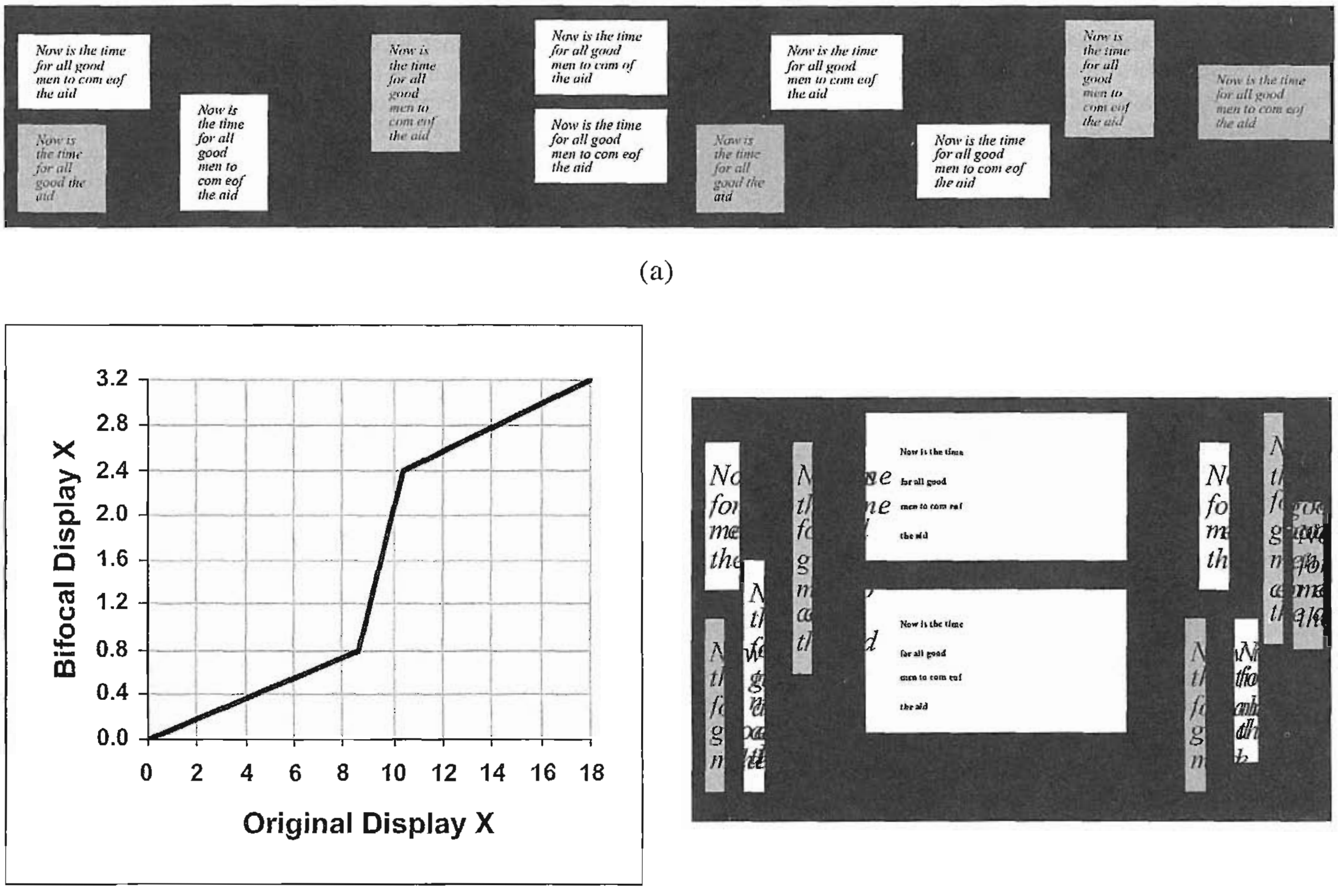


FIGURE 26.26. Micro-macro reading. (Winfree, 1987). Courtesy Scientific American Library.



(a)

(b)

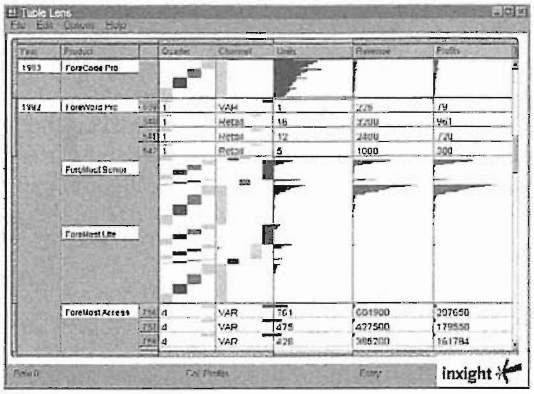
(c)

FIGURE 26.27. Bifocal + transfer function.

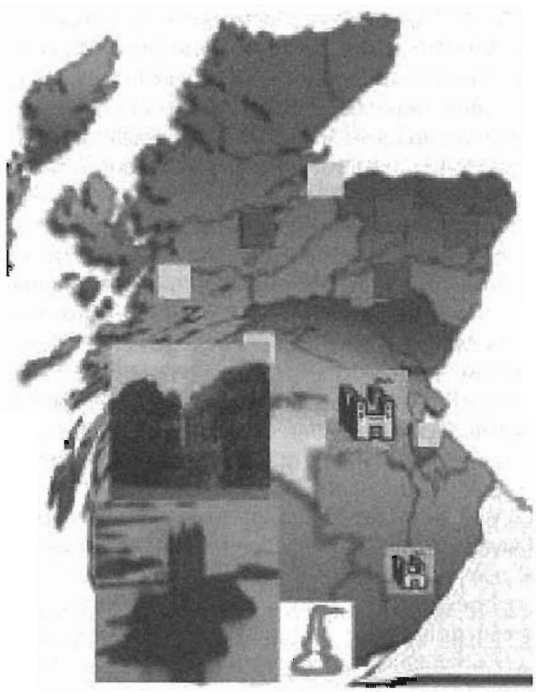
axis in the original display are mapped into how many units in the resultant display. The result could be compression or enlargement of a section of the display. As a result of applying this visual transfer function to Fig. 26.27(a), the display is compressed to Fig. 26.27(d). Actually, the documents in the compressed region have been further altered by using a semantic zooming function to give them a simplified visual form. The form of Fig. 26.27(c) shows that this is essentially a step function of two different slopes. An example of a two-dimensional step function is the Table Lens (Fig. 26.28[a]). The Table Lens is a spreadsheet in which the columns of selected cells are expanded to full size in X and the rows of selected cells are expanded to full size in Y. All other cells are compressed, and their content represented only by a graphic. As a consequence, spreadsheets up to a couple orders of magnitude larger can be represented.

By varying the visual transfer function (see, for example, the review by Leung and Apperley (1994), a wide variety of distorted views can be generated. Figure 26.28(b) shows an application in which a visual transfer function is used to expand a bubble around a local region on a map. The expanded space in the region is used to show additional information about that region. Distorted views must be designed carefully so as not to damage important visual relationships. Bubble distortions of maps may change whether roads appear parallel to each other. However, distorted views can be designed with "flat" and "transition" regions to address this problem. Figure 26.27(a) does not have curvilinear distortions. Focus+context visualizations can be used as part of compact user controls. Keahey (2001) has created an interactive scheme in which the bubble is used to "preview" a region. When the user releases a button over the region, the system zooms in far enough to flatten out the bubble. Bederson has developed a focus+context pull-down menu (Bederson, 2000) that allows the viewing and selection of large lists of typefaces in text editor Fig. 26.27(c).

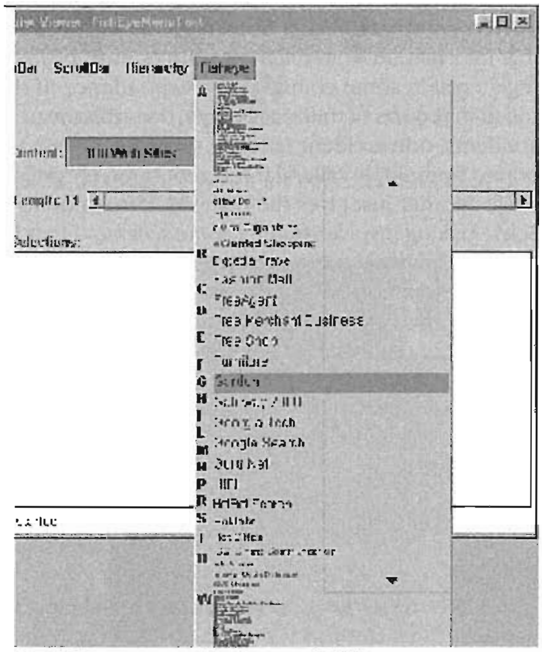
Perspective distortion. One interesting form of distorting visual transfer functions is 3D perspective. Although it can be described with a 2D distorting visual transfer function, it is usually not experienced as distorting by users due to the special perceptual mechanisms humans have for processing 3D. Figure 26.28(c) shows the Perspective Wall (Mackinlay, Robertson, & Card, 1991). Touching any place on the walls animates its transition into the central focal area. The user perceives the



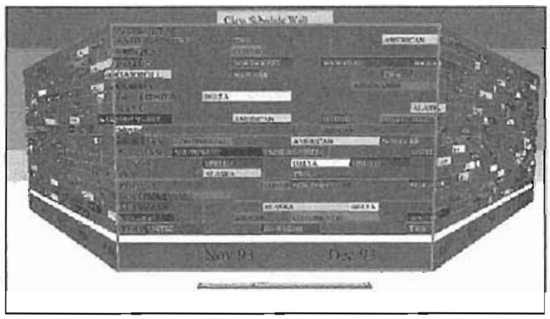
(a) Table Lens. Courtesy of Inxight Software.



(b) Nonlinear distortion of UK.. Courtesy Alan Keahey



(c) Fisheye menus (Bederson, 2000).



(d) Perspective Wall (Mackinlay, Robertson, & Card, 1991).

FIGURE 26.28. Attention-Reactive Visualizations.

context area of the wall as an undistorted 2D image in a 3D space, rather than as a distorted 2D image; however, the same sort of compression is still achieved in the nonfocus area.

Alternate geometries. Instead of altering the size of components, focus+context effects can also be achieved by changing the geometry of the spatial substrate itself. One example is the hyperbolic tree (Lamping & Rao, 1994). A visualization such as a tree is laid out in hyperbolic space (which itself expands exponentially, just like the tree does), and then projected on to the Euclidean plane. The result is that the tree seems to expand around the focal nodes and to be compressed elsewhere. Selecting another node in the tree animates that portion to the focal area. Munzner (Munzner & Burchard, 1995) has extended this notion to 3D hyperbolic trees and used them to visualize portions of the Internet.

SENSEMAKING WITH VISUALIZATION

Knowledge Crystallization

The purpose of information visualization is to amplify cognitive performance, not just to create interesting pictures. Information visualizations should do for the mind what automobiles do for the feet. So here, we return to the higher level cognitive operations of which information visualization is a means and a component. A recurrent pattern of cognitive activity to which information visualization would be useful (though not the only one!) is "knowledge crystallization." In knowledge crystallization tasks, there is a goal (sometimes ill-structured) that requires the acquisition and making sense of a body of information, as well as the creative formulation of a knowledge product, decision, or action. Examples would be writing a scientific paper, business or military intelligence, weather forecasting, or buying a laptop computer. For these tasks, there is usually a concrete outcome of the task—the submitted manuscript of a paper, a delivered briefing, or a purchase. Knowledge crystallization does have characteristic processes, however, and it is by amplifying these that information visualization seeks to intervene and amplify the user's cognitive powers. Understanding of this process is still tentative, but the basic parts can be outlined:

Acquire information. Make sense of it. Create something new. Act on it.

In Table 26.10, we have listed some of the more detailed activities these entail. We can see examples of these in our initial examples.

Acquire information. The FilmFinder is concentrated largely on acquiring information about films. *Search* is one of the methods of acquiring information in Table 26.10, and the FilmFinder is an instance of the use of information visualization in search. In fact, Shneiderman (Card et al., 1999) has identified a heuristic for designing such systems:

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

The user starts with an overview of the films, and then uses sliders to filter the movies, causing the overview to zoom in on the remaining films. Popping up a box gives details on the particular films. The user could use this system as part of a knowledge crystallization process, but the other activities would take place outside the system. The SmartMoney system also uses the TreeMap visualization for acquiring information, but this time the system is oriented toward *monitoring,* another of the methods in Table 26.10. A glance at the sort of chart in Fig. 26.5 allows an experienced user to notice interesting trends among the hundreds of stocks and industries monitored. Another method of acquiring information, *capture*, refers to acquiring information that is tacit or implicit. For example, when users browse the World Wide Web, their paths contain information about their goals. This information can be captured in logs, analyzed, and visualized (Chi & Card, 1999). It is worth making the point that acquiring information is not something that the user must necessarily do explicitly. Search, monitoring, and capture can be implicitly triggered by the system.

TABLE 26.10. Knowledge Crystallization Operators

Acquire Information	Monitor
	Search,
	Capture (make implicit knowledge explicit)
Make sense of it	Extract information
	Fuse different sources
	Find schema
	Recode information into schema
Create something new	Organize for creation
	Author
Act on it	Distribute
	Apply
	Act

Make sense of it. The heart of knowledge crystallization is sensemaking. This process is by no means as mysterious as it might appear. Because sensemaking involving large amounts of information must be externalized, the costs of finding, organizing, and moving information around have a major impact on its effectiveness. The actions of sensemaking itself can be analyzed. One process is *extraction.* Information must be got out of its sources. In our hotel example, the hotel manager extracted information from hotel records. A more subtle issue is that information from different sources must be *fused*—that is, registered in some common correspondence. If there are six called-in reports of traffic accidents, does this mean six different accidents, one accident called in six times, or two accidents reported by multiple callers? If one report merely gives the county, while another just gives the highway, it may not be easy to tell. Sensemaking involves finding some *schema*—that is, some descriptive language—in terms of which information can be compactly expressed (Russell, Stefik, Pirolli, & Card, 1993). In our hotel example, permuting the matrices brought patterns to the attention of the manager. These patterns formed a schema she used to organize and represent hotel stays compactly. In the case of buying a laptop computer, the schema may be a table of features by models. Having a common schema then permits compact description. Instances are *recoded* into the schema. Residual information that does not fit the schema is noted and can be used to adjust the schema.

Create something new. Using the schema, information can be reorganized to create something new. It must be *organized* into a form suitable for the output product and that product must be *authored.* In the case of the hotel example, the manager created the presentation of Fig. 26.7(c).

Act on it. Finally, there is some consequential output of the knowledge crystallization task. That action may be to distribute a report or give a briefing, to act directly in some way, such as setting up a new promotion program for the hotel or buying a laptop on the basis of the analysis, or by giving directives to an organization.

Levels for Applying Information Visualization

Information visualization can be applied to facilitate the various subprocesses of knowledge crystallization just described. It can also be applied at different architectural levels in a system. These have been depicted in Fig. 26.29. At one level is the use of visualization to help users access information outside the im-

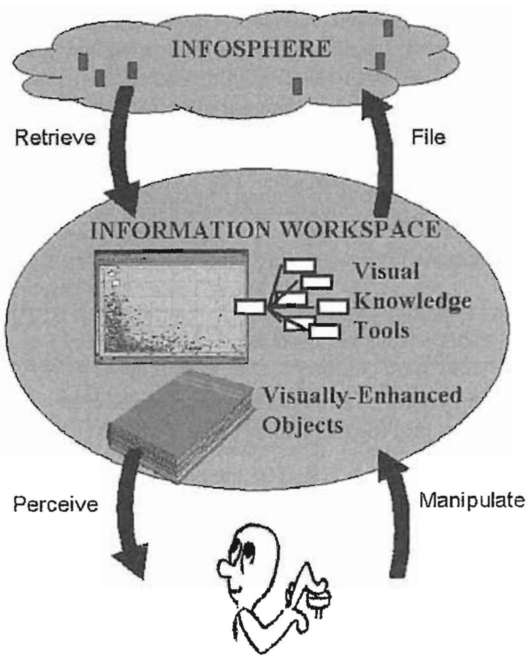


FIGURE 26.29. Levels of use for information visualization.

mediate environment—the *infosphere*—such as information on the Internet or from corporate digital libraries. Figure 26.30(a) shows such a visualization of the Internet (Bray, 1996). websites are laid out in a space such that sites closer to each other in the visualization tend to have more traffic. The size of the disk represents the number of pages in the site. The globe size represents the number of out-links. The globe height shows the number of in-links.

The second level is the *information workspace*. The information workspace is like a desk or workbench. It is a staging area for the integration of information from different sources. An information workspace might contain several visualizations related to one or several tasks. Part of the purpose of an information workspace is to make the cost of access low for information in active use. Figure 26.30(b) shows a 3D workspace for the Internet, the Web Forager (Card, Robertson, & York, 1996). Pages from the World Wide Web, accessed by users through clicking on URLs or searches, appear in the space. These can be organized into piles or books related to different topics. Figure 26.30 (c) shows another document workspace, STARLIGHT (Risch et al., 1997). Documents are represented as galaxies of points in space such that similar documents are near each other. In the workspace, various tools allow linking the documents to maps and other information and analytical resources.

(a) Infosphere: (Bray, 1996).

(b) Workspace: Web Forager (Card, Robertson, & York, 1996).

(c) Workspace: STARLIGHT: (Risch et al., 1997).

(d) Visually-enhanced object: Voxel-Man. Courtesy of University of Hamburg.

FIGURE 26.30. Information visualization applications.

The third level is *visual knowledge tools*. These are tools that allow schema forming and rerepresentation of information. The permutation matrices in Fig. 26.7, the SeeSoft system for analyzing software in Fig. 26.15(b), and the Table Lens in Fig. 26.27(a) are examples of visual knowledge tools. The focus is on determining and extracting the relationships.

The final level is *visually enhanced objects*, coherent information objects enhanced by the addition of information visualization techniques. An example is Fig. 26.30(d), in which voxel data of the brain have been enhanced through automatic surface rendition, coloring, slicing, and labeling. Abstract data structures representing neural projects and anatomical labels have been integrated into a display of the data. Visually enhanced objects focus on revealing more information from some object of intrinsic visual form.

Information visualization is a set of technologies that use visual computing to amplify human cognition with abstract information. The future of this field will depend on the uses to which it is put and how much advantage it gives to these. Information visualization promises to help us speed our understanding and action in a world of increasing information volumes. It is a core part of a new technology of human interfaces to networks of devices, data, and documents.

ACKNOWLEDGMENT

This chapter benefited from detailed and generous discussion with Jock Mackinlay and tutorial collaborations with Ed Chi of PARC.

References

Ahlberg, C., & Shneiderman, B. (1994a). *Visual information seeking using the filmfinder.* Paper presented at the Conference Companion of CHI'94, ACM Conference on Human Factors in Computing Systems.

Ahlberg, C., & Shneiderman, B. (1994b). *Visual information seeking: Tight coupling of dynamic query filters with starfield displays.* Paper presented at the Proceedings of CHI'94, ACM Conference on Human Factors in Computing Systems, New York.

Battista, G. D., Eades, P., Tamassia, R., & Tollis, I. G. (1994). Annotated bibliography on graph drawing. *Computational Geometry: Theory and Applications, 4*(5), 235–282.

Becker, R. A., Eick, S. G., & Wilks, A. R. (1995, March 1). Visualizing network data. *IEEE Transactions on Visualization and Computer Graphics, 1*, 16–28.

Bederson, B. B. (2000). *Fisheye menus.* Paper presented at the UIST 2000, ACM Symposium on User Interface Software and Technology (CHI Letters, 2(2)).

Bertin, J. (1983). *Semiology of graphics: Diagrams, networks, maps* (W. J. Berg, Trans.). Madison, WI: University of Wisconsin Press. (Original work published 1967)

Bertin, J. (1981). Graphic constructions (P. Scott, Trans.). In *Graphics constructions and graphic information-processing* (pp. 24–31). Berlin, Germany: Walter De Gruyter. (Original work published 1977)

Bray, T. (1996). Measuring the web. *Computer Networks and ISDN Systems, 28*(7–11-May), 992.

Card, S. K., Mackinlay, J. D., & Shneiderman, B. (1999). *Information visualization: Using vision to think.* San Francisco: Morgan Kaufmann Publishers.

Card, S. K., Moran, T. P., & Newell, A. (1986). The model human processor: An engineering model of human performance. In J. Thomas (Ed.), *Handbook of perception and human performance* (pp. 41–35). New York: John Wiley and Sons.

Card, S. K., Robertson, G. G., & York, W. (1996). *The webbook and the web forager: An information workspace for the world-wide web.* Paper presented at the Proceedings of CHI'96, ACM Conference on Human Factors in Computing Systems, New York.

Casner, S. (1991, April 2). Task-analytic approach to the automated design of graphic presentations. *ACM Transactions on Graphics, 10*, 111–151.

Chi, E. H., & Card, S. K. (1999). *Sensemaking of evolving websites using visualization spreadsheets.* Paper presented at the Infovis 1999, IEEE Conference on Information Visualization 1999, San Francisco.

Chi, E. H., & Riedl, J. T. (1998). An operator interaction framework for visualization spreadsheets. Paper presented at the *Proceedings of InfoVis'98, IEEE Symposium on Information Visualization.*

Chuah, M. C., Roth, S. F., Mattis, J., & Kolojejchick, J. A. (1995). Sdm: Malleable information graphics. Paper presented at the *Proceedings of InfoVis'95, IEEE Symposium on Information Visualization*, New York.

Cleveland, W. S., & McGill, M. E. (1988). *Dynamic graphics for statistics.* Pacific Grove, California: Wadsworth and Brooks/Cole.

Cruz, I. F., & Tamassia, R. (1998). *Graph drawing tutorial.* Retrieved (date, from http://www.cs.brown.edu/people/rt/papers/gd-tutorial/gd-constraints.pdf.

Eick, S. G., Steffen, J. L., & Sumner, E. E. (1992, November). Seesoft—a tool for visualizing software. *IEEE Transactions on Software Engineering, 18*(11), 957–968.

Eick, S. G., & Wills, G. J. (1993, October 25–29). Navigating large networks with hierarchies. Paper presented at the *Proceedings of IEEE Visualization'93 Conference*, San Jose, CA.

Engelhardt, Y., Bruin, J. D., Janssen, T., & Scha, R. (1996). The visual grammar of information graphics. In S. University (Ed.), *Artificial intelligence in design workshop notes* (pp. 24–27).

Fairchild, K. M., Poltrock, S. E., & Furnas, G. W. (1988). Semnet: Three-dimensional representations of large knowledge bases. In R. Guindon (Ed.), *Cognitive science and its applications for human-computer interaction* (pp. 201–233). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Feiner, S., & Beshers, C. (1990). Worlds within worlds: Metaphors for exploring n-dimensional virtual worlds. In *ACM symposium on User Interface Software.*

Freeman, E., & Fertig, S. (1995). Lifestreams: Organizing your electronic life. Paper presented at the *Proceedings of AAAI Fall Symposium on AI Applications in Knowledge Navigation.*

Furnas, G. W. (1981). The fisheye view: A new look at structured files. In B. Shneiderman (Ed.), *Readings in information visualization: Using vision to think* (pp. 312–330). San Francisco: Morgan Kaufmann Publishers, Inc.

Healey, C. G., Booth, K. S., & Enns, J. T. (1995). High-speed visual estimation using preattentive processing. *ACM Transactions on Computer-Human Interaction, 3*(2), 107–135.

Inselberg, A. (1997). Multidimensional detective. Paper presented at the *Proceedings of InfoVis'97, IEEE Symposium on Information Visualization, IEEE Information Visualization.*

Inselberg, A., & Dimsdale, B. (1990). Parallel coordinates : A tool for visualizing multi-dimensional geometry. Paper presented at the *Proceedings of IEEE Visualization'90 Conference*, Los Alamitos, CA.

Keahey, T. A. (2001, October 22–23, 2001). *Getting along: Composition of visualization paradigms.* Paper presented at the Infovis 2001, IEEE Information Visualization 2001, San Diego, California.

Keim, D. A., & Kriegel, H.-P. (1994, September). Visdb: Database exploration using multidimensional visualization. *IEEE Computer Graphics and Applications*, 40–49.

Kosslyn, S. M. (1994). *Image and brain: The resolution of the imagery debate*. Cambridge, MA: The MIT PRess.

Lamping, J., & Rao, R. (1994). *Laying out and visualizing large trees using a hyperbolic space.* Paper presented at the Proceedings of UIST'94, ACM Symposium on User Interface Software and Technology.

Leung, Y. K., & Apperley, M. D. (1994, June). A review and taxonomy of distortion-orientation presentation techniques. *ACM Transactions on Computer-Human Interaction, 1*(2), 126–160.

Lin, X., Soergel, D., & Marchionini, G. (1991). *A self-organizing semantic map for information retrieval.* Paper presented at the Proceedings of SIGIR'91, ACM Conference on Research and Development in Information Retrieval, Chicago, IL.

MacEachren, A. M. (1995). *How maps work*. New York: The Guilford Press.

Mackinlay, J. D. (1986a). *Automatic design of graphical presentations.* Unpublished doctoral dissertation, Stanford University, California.

Mackinlay, J. D. (1986b). Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics, 5*(2), 110–141.

Mackinlay, J. D., Robertson, G. G., & Card, S. K. (1991). *The perspective wall: Detail and context smoothly integrated.* Paper presented at the Proceedings of CHI'91, ACM Conference on Human Factors in Computing Systems, New York.

McCormick, B. H., & DeFanti, T. A. (1987, November 6). Visualization is scientific computing. *Computer Graphics, 21*.

Munzner, T., & Burchard, P. (1995, December 14–15). *Visualizing the structure of the world wide web in 3d hyperbolic space.* Paper presented at the Proceedings of VRML '95.

Playfair, W. (1786). *The commercial and political atlas*. London.

Resnikoff, H. L. (1989). *The illusion of reality*. New York: Springer-Verlag.

Risch, J. S., Rex, D. B., Dowson, S. T., Walters, T. B., May, R. A., & Moon, B. D. (1997). *The starlight information visualization system.* Paper presented at the Proceedings of IEEE International Conference on Information Visualization, London, England.

Robertson, G. G., Card, S. K., & Mackinlay, J. D. (1989). *The cognitive coprocessor for interactive user interfaces.* Paper presented at the Proceedings of UIST'89, ACM Symposium on User Interface Software and Technology.

Robertson, G. G., Card, S. K., & Mackinlay, J. D. (1993). Information visualization using 3d interactive animation. *Communications of the ACM, 36*(4), 57–71.

Robin, H. (1992). *The scientific image: From cave to computer*. New York: H. N. Abrams, Inc.

Roth, S. F., & Mattis, J. (1990). *Data characterization for intelligent graphics presentation.* Paper presented at the Proceedings of CHI'90, ACM Conference on Human Factors in Computing Systems, New York.

Russell, D. M., Stefik, M. J., Pirolli, P., & Card, S. K. (1993). *The cost structure of sensemaking.* Paper presented at the Proceedings of INTERCHI'93, ACM Conference on Human Factors in Computing Systems, Amsterdam.

Shneiderman, B. (1992). Tree visualization with tree-maps: A 2-dimensional space filling approach. *ACM Transactions on Graphics, 11*(1), 92–99.

Shneiderman, B., & Wattenberg, M. (2001). *Ordered tree layouts.* Paper presented at the IEEE Symposium on Information Visualization, San Diego, California.

Spence, R. (2000). *Information visualization*. Harlow, England: Addison-Wesley.

Spence, R., & Apperley, M. (1982). Data base navigation: An office environment for the professional. *Behavior and Information Technology, 1*(1), 43–54.

Tamassia, R. (1996, December 4). Strategic directions in computational geometry working group report. *ACM Computing Surveys, 28*, 591– 606.

Tufte, E. R. (1983). *The visual display of quantitative information*. Cheshire, CT: Graphics Press.

Tufte, E. R. (1990). *Envisioning information*. Cheshire, CT: Graphics Press.

Tufte, E. R. (1997). *Visual explanations: Images and quantities, evidence and narrative*. Cheshire, CT: Graphics Press.

Tukey, J. W. (1977). *Exploratory data analysis*. Reading, MA: Addison-Wesley.

Tweedie, L. A., Spence, R., Dawkes, H., & Su, H. (1996). *Externalising abstract mathematical models.* Paper presented at the Proceedings of CHI'96, ACM Conference on Human Factors in Computing Systems.

Ware, C. (2000). *Information visualization: Perception for design*. San Francisco: Morgan Kaufmann Publishers.