CHAPTER 6

# Static and Moving Patterns

Data mining is about finding patterns that were previously unknown or that depart from the norm. The stock-market analyst looks for any pattern of variables that may predict a future change in price or earnings. The marketing analyst is interested in perceiving trends and patterns in a customer database. When we look for patterns, we are making visual queries that are key to visual thinking. Sometimes the queries are vague; we are on the lookout for a variety of structures in the data. Sometimes they are precise, as when we look for a positive trend in a graph. In data exploration, seeing a pattern can often lead to a key insight, and this is the most compelling reason for visualization.

What does it take for us to see a group? How can 2D space be divided into perceptually distinct regions? Under what conditions are two patterns recognized as similar? What constitutes a visual connection between objects? These are some of the perceptual questions addressed in this chapter. The answers are central to visualization, because most data displays are two-dimensional and pattern perception deals with the extraction of structure from 2D space.

Consider again our three-stage model of perception (illustrated in Figure 6.1). At the early stages of feature abstraction, the visual image is analyzed in terms of primitive elements of form, motion, color, and stereoscopic depth. At the next 2D pattern perception stage, the contours are discovered and the visual world is segmented into distinct regions, based on texture, color, motion, and contour. Next, the structures of objects and scenes are discovered, using information about the connections between component parts, shape-from-shading information, and so on. Pattern perception can be thought of as a set of mostly 2D processes occurring between feature analysis and full object perception, although aspects of 3D space perception, such as stereoscopic depth and structure-from-motion, can be considered particular kinds of pattern perception. Finally, objects and significant patterns are pulled out by attentional processes to meet the needs of the task at hand.

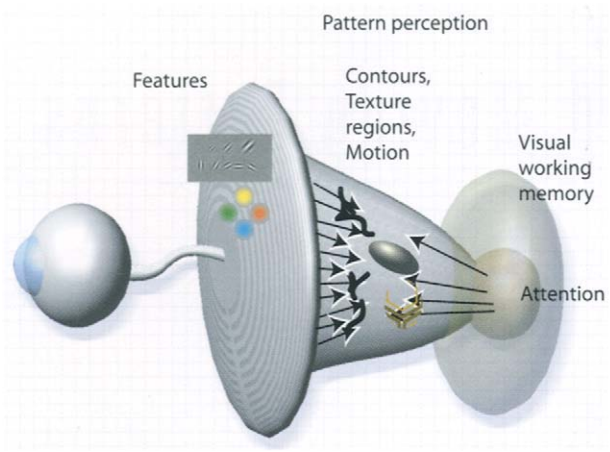



**Figure 6.1** Pattern perception forms a middle ground where the bottom-up processes of feature processing meet the requirements of active attention.

There are radical changes in the kinds of processing that occur at the different stages. In the early stages, massively parallel processing of the entire image occurs. This drives perception from the bottom up. But object and visual search recognition is driven from the top down through active attention, meeting the requirements of visual thinking. At the top level, only three to five objects (or patterns) are held in visual working memory. Pattern perception is the flexible middle ground where objects are extracted from patterns of features. Active processes of attention reach down into the pattern space to keep track of those objects and to analyze them for particular tasks; the essentially bottom-up processing of feature primitives meets the top-down processes of cognitive perception. Rensink (2000) calls the middle ground a "proto-object flux."

Understanding pattern perception provides abstract design rules that can tell us much about how we should organize data so that important structures will be perceived. If we can map information structures to readily perceived patterns, then those structures will be more easily interpreted.

Learning is important in the pattern mechanism. It occurs in the short term through visual priming and in the long term as a kind of skill learning. *Priming* refers to the fact that once we have seen a pattern, it becomes much easier to identify on subsequent appearance. Long-term learning of patterns occurs over hundreds or thousands of trials, but some patterns are much easier to learn than others (Fine and Jacobs, 2002). In this chapter, we consider 2D-pattern perception and what this tells us about information display. In the next two chapters, we consider 3D-space perception, much of which is a form of advanced pattern perception.

# Gestalt Laws

The first serious attempt to understand pattern perception was undertaken by a group of German psychologists who, in 1912, founded what is known as the Gestalt school of psychology. The group consisted principally of Max Westheimer, Kurt Koffka, and Wolfgang Kohler (see Koffka, 1935, for an original text). The word *gestalt* simply means *pattern* in German. The work of the Gestalt psychologists is still valued today because they provided a clear description of many basic perceptual phenomena. They produced a set of *Gestalt laws* of pattern perception. These are robust rules that describe the way we see patterns in visual displays, and although the neural mechanisms proposed by these researchers to explain the laws have not withstood the test of time, the laws themselves have proved to be of enduring value. The Gestalt laws easily translate into a set of design principles for information displays. Eight Gestalt laws are discussed here: proximity, similarity, connectedness, continuity, symmetry, closure, relative size, and common fate (the last concerns motion perception and appears later in the chapter).

## Proximity

Spatial proximity is a powerful perceptual organizing principle and one of the most useful in design. Things that are close together are perceptually grouped together. Figure 6.2 shows two arrays of dots that illustrate the proximity principle. Only a small change in spacing causes us to change what is perceived from a set of rows, in Figure 6.2(a), to a set of columns, in Figure 6.2(b). In Figure 6.2(c), the existence of two groups is perceptually inescapable. Proximity is not the only factor in predicting perceived groups. In Figure 6.3, the dot labeled $x$ is perceived to be part of cluster a rather than cluster b, even though it is as close to the other points in cluster b

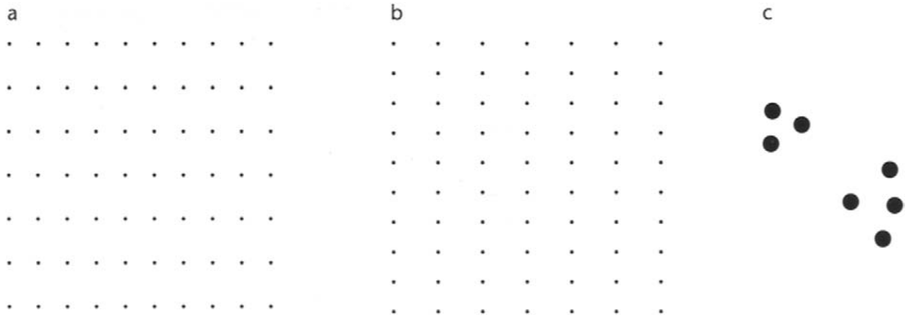



**Figure 6.2** Spatial proximity is a powerful cue for perceptual organization. A matrix of dots is perceived as rows on the left (a) and columns on the right (b). In (c), because of proximity relationships, we perceive two groupings of dots.

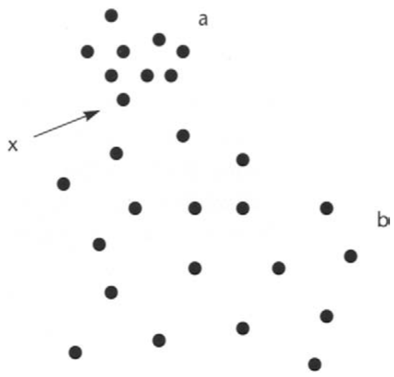



**Figure 6.3** The principle of spatial concentration. The dot labeled *x* is perceived as part of group a rather than group b.

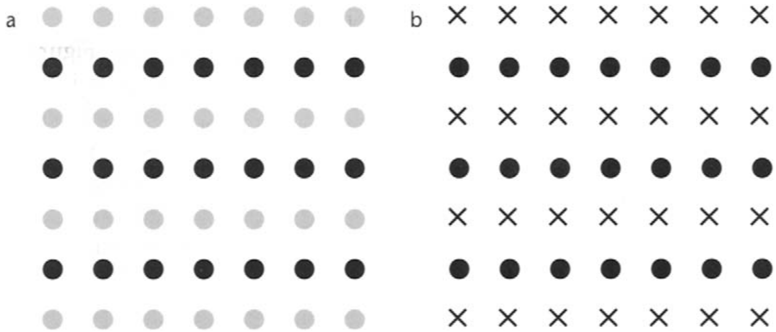



**Figure 6.4** According to the Gestalt psychologists, similarity between the elements in alternate rows causes the row percept to dominate.

as they are to each other. Slocum (1983) called this the *spatial concentration principle*. According to this principle, we perceptually group regions of similar element density.

The application of the proximity law in display design is straightforward: the simplest and most powerful way to emphasize the relationships between different data entities is to place them in proximity in a display.

## Similarity

The shapes of individual pattern elements can also determine how they are grouped. Similar elements tend to be grouped together. In both Figure 6.4(a) and (b), the similarity of the elements causes us to see the rows most clearly.

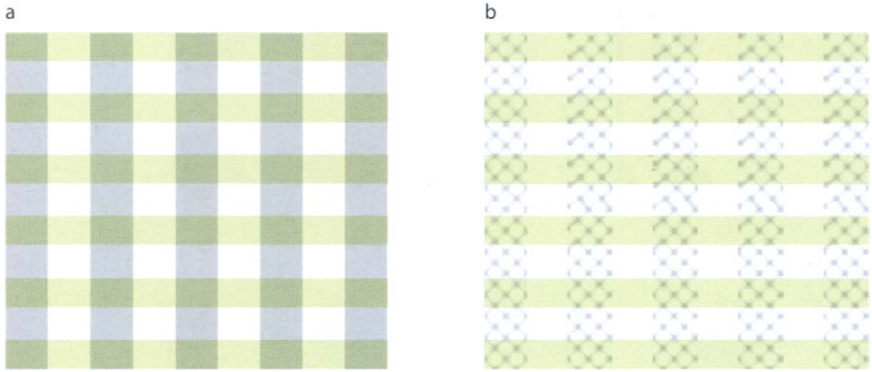



**Figure 6.5** (a) Integral dimensions are used to delineate rows and columns. (b) When separable dimensions (color and texture) are used, it is easier to attend separately to either the rows or the columns.

We can also apply lessons from the concept of integral and separable dimensions that was discussed in Chapter 5. Figure 6.5 shows two different ways of visually separating row and column information. In 6.5(a), integral color and gray-scale coding is used. In Figure 6.5(b), green color is used to delineate rows and texture is used to delineate columns. Color and texture are separable dimensions, and the result is a pattern that can be visually segmented either by rows or by columns. This technique can be useful if we are designing so that users can easily attend to either one pattern or the other.

## Connectedness

Palmer and Rock (1994) argue that connectedness is a fundamental Gestalt organizing principle that the Gestalt psychologists overlooked. The demonstrations in Figure 6.6 show that connectedness can be a more powerful grouping principle than proximity, color, size, or shape. Connecting different graphical objects by lines is a very powerful way of expressing that there is some relationship between them. Indeed, this is fundamental to the node–link diagram, one of the most common methods of representing relationships between concepts.

## Continuity

The Gestalt principle of continuity states that we are more likely to construct visual entities out of visual elements that are smooth and continuous, rather than ones that contain abrupt changes in direction. (See Figure 6.7.)

The principle of good continuity can be applied to the problem of drawing diagrams consisting of networks of nodes and the links between them. It should be easier to identify the sources and destinations of connecting lines if they are smooth and continuous. This point is illustrated in Figure 6.8.

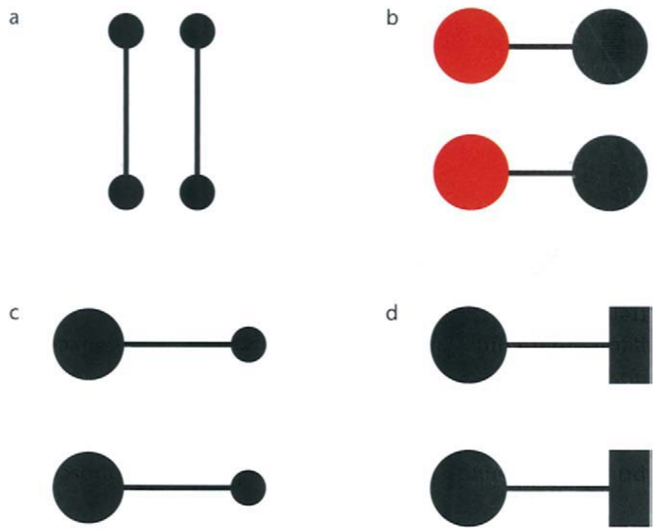



**Figure 6.6** Connectedness is a powerful grouping principle that is stronger than (a) proximity, (b) color, (c) size, or (d) shape.

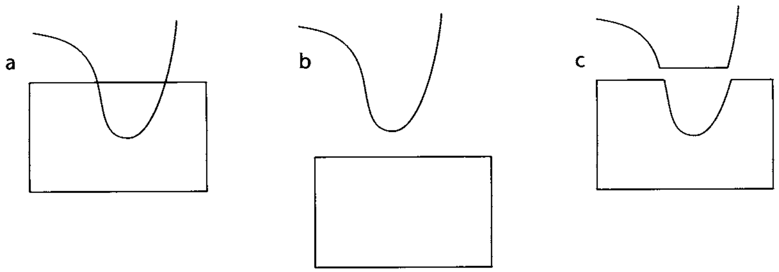



**Figure 6.7** The pattern on the left (a) is perceived as a curved line overlapping a rectangle (b) rather than as the more angular components shown in (c).

## Symmetry

Symmetry can provide a powerful organizing principle. Figures 6.9 and 6.10 provide two examples. The symmetrically arranged pairs of lines in Figure 6.9 are perceived much more strongly as forming a visual whole than the pair of parallel lines. In Figure 6.10(a), symmetry may be the reason why the cross shape is perceived, as opposed to shapes in 6.10(b), even though the second option is not more complicated. A possible application of symmetry is in tasks in which data analysts are looking for similarities between two different sets of time-series data. It may be easier

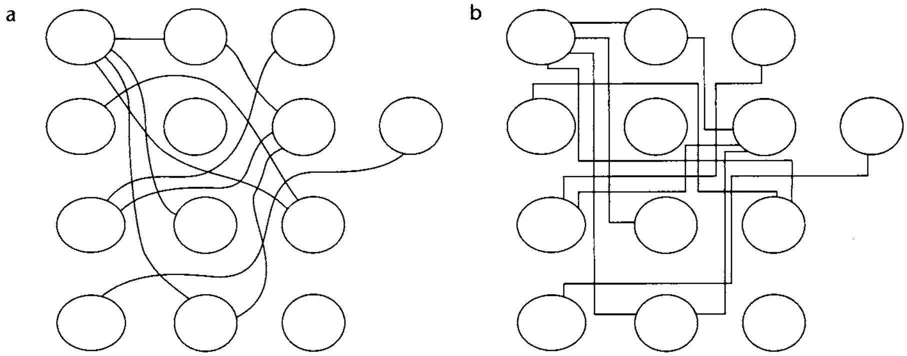



**Figure 6.8** In (a), smooth continuous contours are used to connect the elements, whereas in (b), lines with abrupt changes in direction are used. It is much easier to perceive connections when contours connect smoothly.

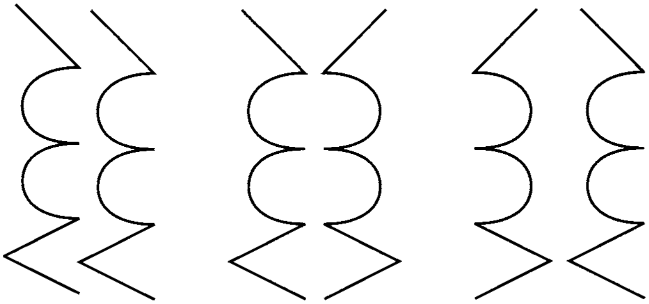


**Figure 6.9** The pattern on the left consists of two identical parallel contours. In each of the other two patterns, one of the contours has been reflected about a vertical axis, producing bilateral symmetry. The result is a much stronger sense of a holistic figure.

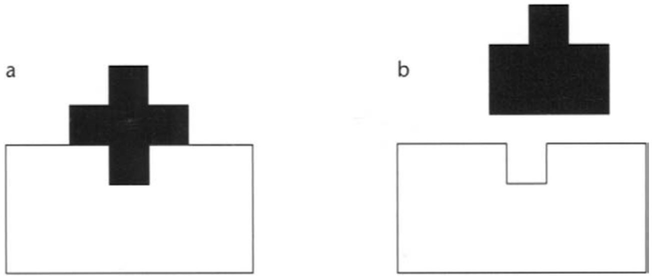



**Figure 6.10** We interpret pattern (a) as a cross in front of a rectangle. An alternative, two objects shown in (b) are not perceived, even though the black shape behind the white shape would be an equally simple interpretation. The cross on the rectangle interpretation has greater symmetry (about horizontal axes) for both of the components.

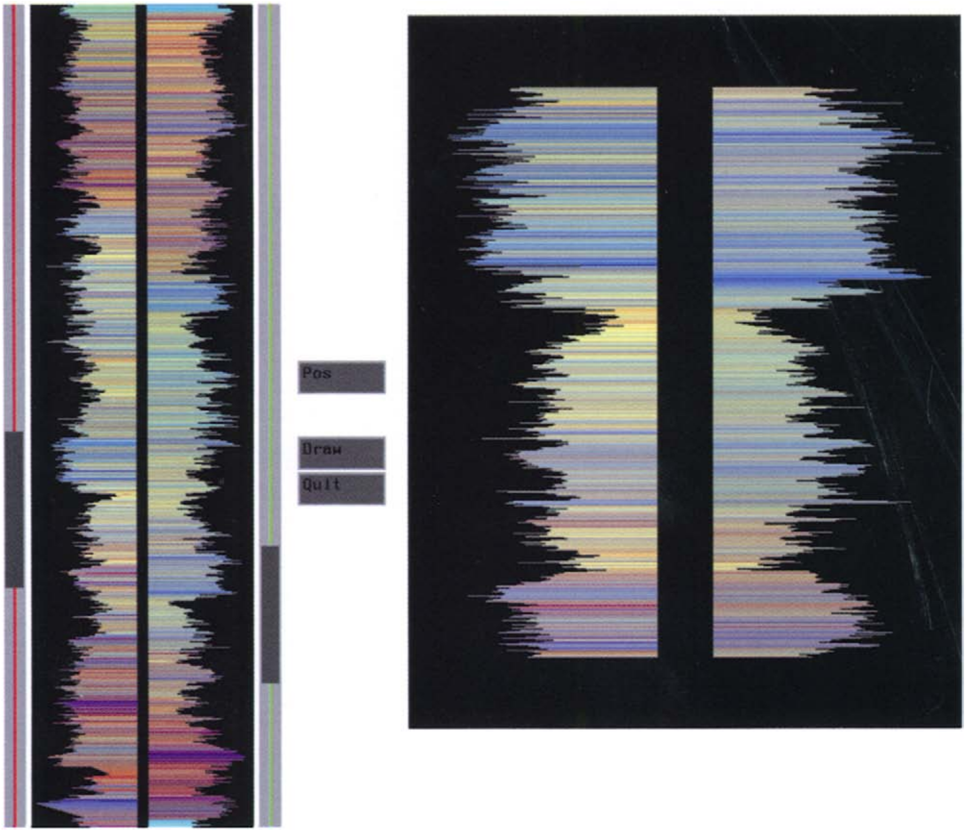



**Figure 6.11** An application designed to allow users to recognize similar patterns in different time-series plots. The data represents a sequence of measurements made on deep ocean drilling cores. Two subsets of the extended sequences are shown on the right.

to perceive similarities if these time series are arranged using vertical symmetry, as shown in Figure 6.11, rather than using the more conventional parallel plots.

## Closure

A closed contour tends to be seen as an object. The Gestalt psychologists argued that there is a perceptual tendency to close contours that have gaps in them. This can help explain why we see Figure 6.12(a) as a complete circle and a rectangle rather than as a circle with a gap in it as in Figure 6.12(b).

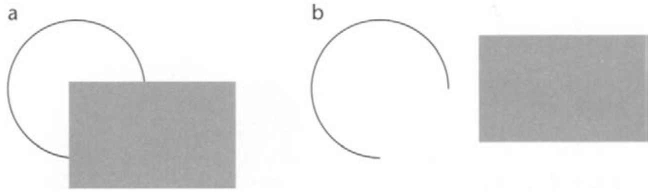



**Figure 6.12** The Gestalt principle of closure holds that neural mechanisms operate to find perceptual solutions involving closed contours. Hence in (a), we see a circle behind a rectangle, not a broken ring as in (b).

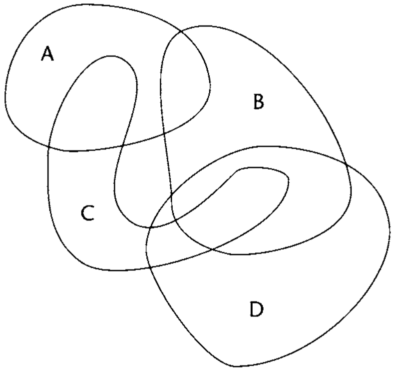



**Figure 6.13** An Euler diagram. This diagram tells us (among other things) that entities can simultaneously be members of sets A and C but not of A, B, and C. Also, anything that is a member of both B and C is also a member of D. These rather difficult concepts are clearly expressed and understood by means of closed contours.

Wherever a closed contour is seen, there is a very strong perceptual tendency to divide regions of space into "inside" or "outside" the contour. A region enclosed by a contour becomes a *common region* in the terminology of Palmer (1992). He showed common region to be a much stronger organizing principle than simple proximity. This, presumably, is the reason why Venn-Euler diagrams are such a powerful device for displaying the interrelationships among sets of data. In an Euler diagram, we interpret the region inside a closed contour as defining a set of elements. Multiple closed contours are used to delineate the overlapping relationships among different sets. A Venn diagram is a more restricted form of Euler diagram containing all possible regions of overlap. The two most important perceptual factors in this kind of diagram are closure and continuity.

A fairly complex structure of overlapping sets is illustrated in Figure 6.13, using an Euler diagram. This kind of diagram is almost always used in teaching introductory set theory, and this in itself is evidence for its effectiveness. Students easily understand the diagrams, and they

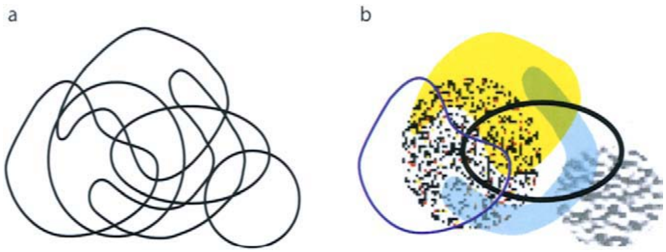



**Figure 6.14** An Euler diagram enhanced using texture and color can convey a more complex set of relations than a conventional Euler diagram using only closed contour.

can transfer this understanding to the more difficult formal notation. Stenning and Oberlander (1994) theorize that the ease with which Euler diagrams can be understood results specifically from the fact that they have limited expressive power, unlike fully abstract formal notation.

Although simple contours are generally used in Euler diagrams to show set membership, we can effectively define regions using color and texture as well, as discussed in Chapters 4 and 5. Indeed, by using both we should be able to create Euler diagrams that are considerably more complex and still readily understandable. Figure 6.14 illustrates.

Closed contours are extremely important in segmenting the monitor screen in windows-based interfaces. The rectangular overlapping boxes provide a strong segmentation cue, dividing the display into different regions. In addition, rectangular frames provide frames of reference: the position of every object within the frame tends to be judged relative to the enclosing frame. (See Figure 6.15.)

## Relative Size

In general, smaller components of a pattern tend to be perceived as objects. In Figure 6.16, a black propeller is seen on a white background, as opposed to the white areas being perceived as objects.

## Figure and Ground

Gestalt psychologists were also interested in what they called *figure–ground* effects. A *figure* is something objectlike that is perceived as being in the foreground. The *ground* is whatever lies behind the figure. The perception of figure as opposed to ground can be thought of as the fundamental perceptual act of identifying objects. All the Gestalt laws contribute to creating a figure, along with other factors that the Gestalt psychologists did not consider, such as texture segmentation (see Chapter 5). Closed contour, symmetry, and the surrounding white area all con-

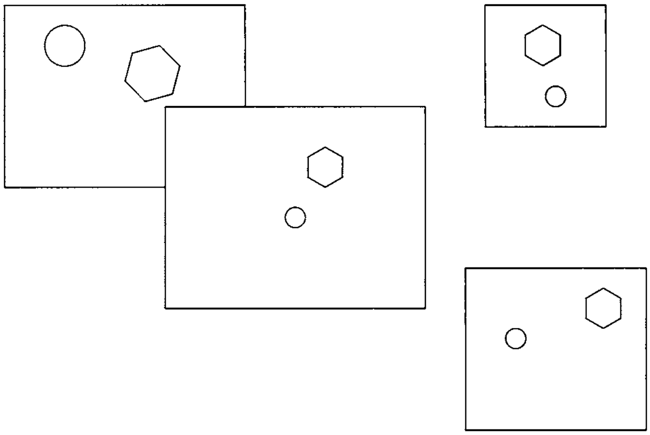


**Figure 6.15** Closed rectangular contours strongly segment the visual field. They also provide reference frames. Both the positions and the sizes of enclosed objects are, to some extent, interpreted with respect to the surrounding frame.

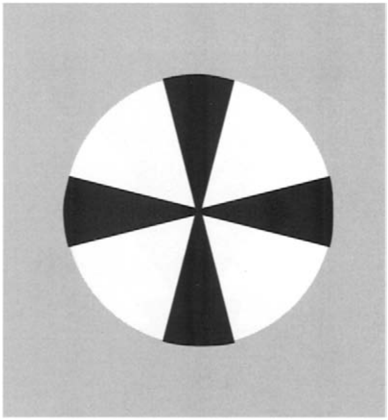


**Figure 6.16** The black areas are smaller, and therefore more likely to be perceived as an object. It is also easier to perceive patterns that are oriented horizontally and vertically as objects.

tribute to the perception of the shape in Figure 6.17 as figure, as opposed to a cut-out hole, for example.

Figure 6.18 shows the classic Rubin's Vase figure, in which it is possible to perceive either two faces, nose to nose, or a black vase centered in the display. The fact that the two percepts tend to alternate suggests that competing active processes may be involved in trying to construct figures from the pattern.

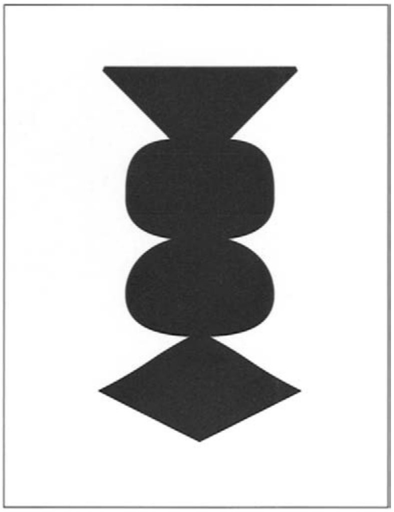


**Figure 6.17** Symmetry, surrounding white space, and a closed contour all contribute to the strong sense that this shape is figure, rather than ground.

**Figure 6.18** Rubin's Vase. The cues for figure and ground are roughly equally balanced, resulting in a bistable percept of either two faces or a vase.

# More on Contours

A contour is a continuous perceived boundary between regions of a visual image. A contour can be defined by a line, by a boundary between regions of different color, by stereoscopic depth, by motion patterns, or by texture. Contours can even be perceived where there are none. Figure 6.19 illustrates an *illusory contour*; a ghostly boundary of a blobby shape is seen even where

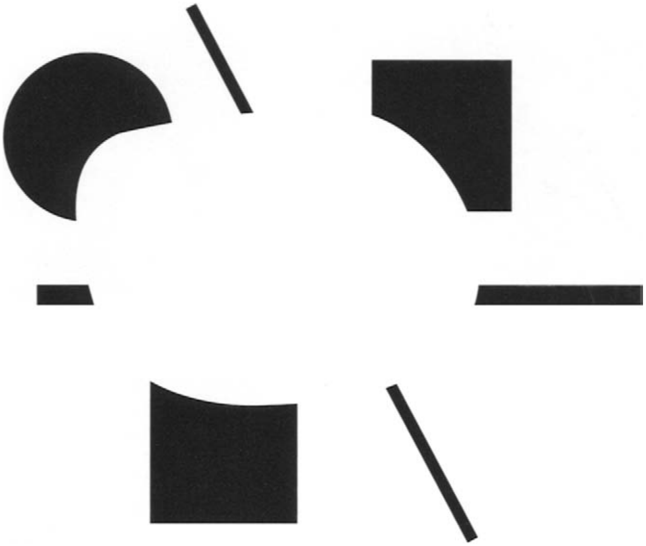


**Figure 6.19** Illusory contour.

none is physically present. There is extensive literature on illusory contours (see Kanizsa, 1976, for an early review).

Because the process that leads to the identification of contours is seen as fundamental to object perception, contour detection has received considerable attention from vision researchers. There are a number of detailed neurophysiological models designed to explain how contours can be extracted from the visual image, based on what is known about early visual processing. See Marr (1982), for example.

Higher-order neurophysiological mechanisms of contour perception are not well understood. However, one result is intriguing. Gray et al. (1989) found that cells with collinear receptive fields tend to fire in synchrony. Thus, we do not need to propose higher-order feature detectors, responding to more and more complex curves, to understand the neural encoding of contour information. Instead, it may be that groups of cells firing in synchrony is the way that the brain holds related pattern elements in mind. Theorists have suggested a fast enabling link, a kind of rapid feedback system, to achieve the firing of cells in synchrony. For a review, see Singer and Gray (1995).

Fortunately, because a theoretical understanding is only just emerging, the exact mechanisms involved in contour detection are less relevant to the purpose of designing visualizations than are the circumstances under which we perceive contours. A set of experiments by Field et al. (1993) places the Gestalt notion of *good continuation* on a firmer scientific basis. In these experiments, subjects had to detect the presence of a continuous path in a field of 256 randomly oriented Gabor patches (see Chapter 5 for a discussion of Gabor functions). The setup is illustrated

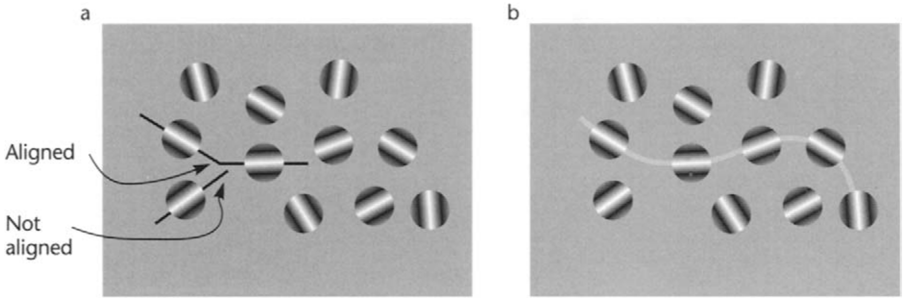



**Figure 6.20** A schematic diagram illustrating the experiments conducted by Field et al. (1993). If the elements were aligned as shown in (a) so that a smooth curve could be drawn through some of them, the curve shown in (b) was perceived. In the actual experiments, Gabor patches were used.

schematically in Figure 6.20. The results show that subjects were very good at perceiving a smooth path through a sequence of patches. As one might expect, continuity between Gabor patches oriented in straight lines was the easiest to perceive. More interesting, even quite wiggly paths were readily seen if the Gabor elements were aligned as shown in Figure 6.20(b).

There are direct applications of this result in displaying vector field data. A common technique is to create a regular grid of oriented arrows, such as the one shown in Figure 6.21. When the arrows are displaced so that smooth contours can be drawn between them, the flow pattern is much easier to see.

# Perceiving Direction: Representing Vector Fields

The perception of contour leads us naturally to the perceptual problem of representing vector fields. This problem can be broken down into two components: the representation of orientation and the representation of magnitude. Some techniques display one component but not both.

Instead of using little arrows, one obvious and effective way of representing vector fields is through the use of continuous contours; a number of effective algorithms exist for this purpose. Figure 6.22 shows an example from Turk and Banks (1996). This effectively illustrates the direction of the vector field, although it is ambiguous in the sense that for a given contour there can be two directions of flow. Conventional arrowheads can be added, as in Figure 6.21, but the result is visual clutter. In addition, in Figure 6.22 the magnitudes of the vectors are given by line density and inverse line width, and this is not easy to read.

An interesting way to resolve the flow direction ambiguity is provided in a seventeenth-century vector field map of North Atlantic wind patterns by Edmund Halley (discussed in Tufte, 1983). Halley's elegant pen strokes, illustrated in Figure 6.23, are shaped like long, narrow airfoils oriented to the flow, with the wind direction given by the blunt end. Interestingly, Halley also arranges his strokes along streamlines. We verified experimentally that strokes like Halley's are unambiguously interpreted with regard to direction (Fowler and Ware, 1989).

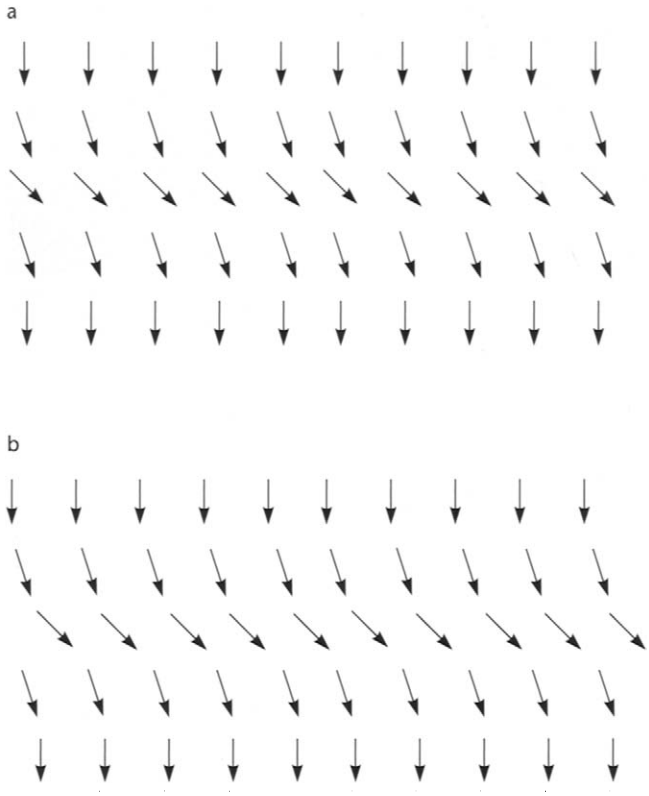



**Figure 6.21** The results of Field et al. (1993) suggest that vector fields should be easier to perceive if smooth contours can be drawn through the arrows. (a) A regular grid is used to determine arrow layout. (b) The arrows have been shifted so that smooth contours can be drawn through the arrows. As theory predicts, the latter is more effective.

We also developed a new method for creating an unambiguous sense of vector field direction that involves varying the color along the length of a stroke. This is illustrated in Figure 6.24. There was a strong interaction between the direction of color change and the background color. If one end of the stroke was given the background color, the stroke direction was perceived to be in the direction of color change away from the background color. In our experiments, the impression of direction produced by color change completely dominated that given by shape.

## Comparing 2D Flow Visualization Techniques

Laidlaw et al. (2001) carried out an experimental comparison of the six different flow visualization methods illustrated in Figure 6.25 and briefly described as follows.

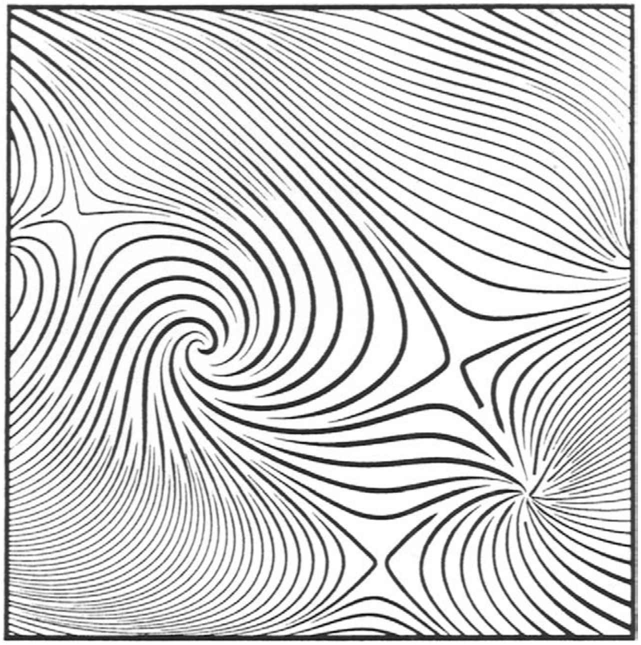


**Figure 6.22** Vector field streamlines are an effective way to represent vector field or flow field data. However, the direction is ambiguous and the magnitude is not clearly expressed (Turk and Banks, 1996).

**Figure 6.23** Drawing in a style based on the pen strokes used by Edmund Halley (1696), discussed in Tufte (1983), to represent the trade winds of the North Atlantic. Halley described the wind direction as being given by "the sharp end of each little stroak pointing out that part of the horizon, from whence the wind continually comes."

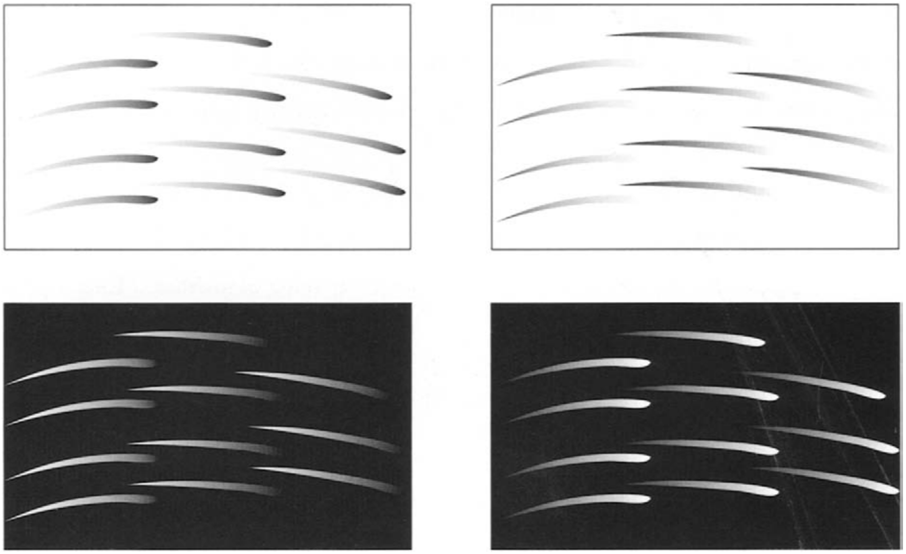


**Figure 6.24** Vector direction can be unambiguously given by means of color change relative to the background.

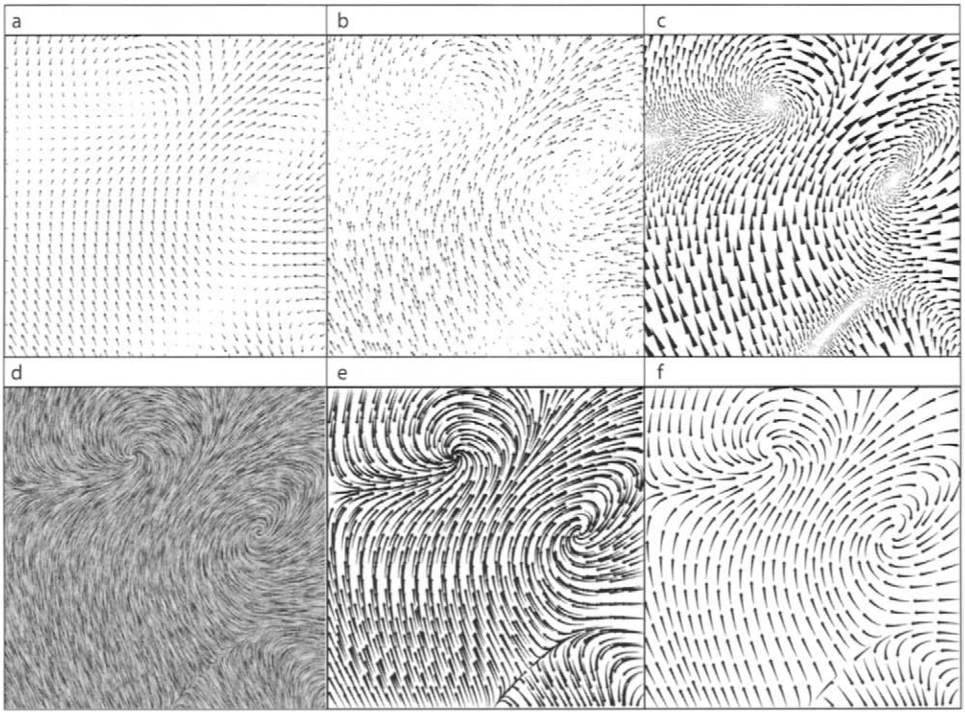



**Figure 6.25** Six different flow visualization techniques evaluated by Laidlaw et al., 2001. *Used by permission.*

(a) Arrows on a regular grid. Fixed length.

(b) Arrows on a jittered grid to reduce perceptual aliasing effects. Fixed length.

(c) Triangle icons. Icon size proportional to field strength density inversely related to icon size (Kirby et al., 1999).

(d) Line integral convolution (Cabral and Leedom, 1993).

(e) Large-head arrows along a streamline using a regular grid (Turk and Banks, 1996).

(f) Large-head arrows along streamlines using constant spacing algorithm. (Turk and Banks, 1996).

In order to evaluate any visualization, it is necessary to specify a set of tasks. Laidlaw et al. (2001) had subjects identify critical points as one task. These are points in a vector or flow field where the vectors have zero magnitude. The results showed the arrow-based methods illustrated in Figure 6.25(a) and (b) to be the least effective for identifying the locations of these points. A second task involved perceiving advection trajectories. An *advection trajectory* is the path taken by a particle dropped in a flow. The streamline methods of Turk and Banks proved best for showing advection, especially the method shown in Figure 6.25(f). The line integral convolution method, shown in Figure 6.25(d), was by far the worst for advection, probably because it does not unambiguously identify direction.

Although the study done by Laidlaw et al. (2001) is the first serious comparative evaluation of the effectiveness of vector field visualization methods, it is by no means exhaustive. There are alternative visualizations, and those shown have many possible variations: longer and shorter line segments, color variations, and so on. In addition, the tasks studied by Laidlaw et al. do not include all of the important visualization tasks that are likely to be carried out with flow visualizations. Here is a more complete list:

- Identifying the location and nature of critical points
- Judging an advection trajectory
- Perceiving patterns of high and low velocity
- Perceiving patterns of high and low vorticity (sometimes called *curl*)
- Perceiving patterns of high and low turbulence

Both the kinds and the scale of patterns that are important will vary from one application to another; small-scale detailed patterns, such as eddies, will be important to one researcher, whereas large-scale patterns will interest another.

The problem of optimizing flow display may not be quite so complex and multifaceted as it would first seem. If we ignore the diverse algorithms and think of the problem in purely visual terms, then the various display methods illustrated in Figures 6.22 through 6.25 have many characteristics in common. They all consist principally of contours oriented in the flow direction, although these contours have different characteristics in terms of length, width, and shape. The line integral convolution method illustrated in Figure 6.25(d) produces a very different-looking, blurry result; however, something similar could be computed using blurred contours. Contours that vary in shape and gray value along their lengths could be expressed with two or three parameters. The different degrees of randomness in the placement of contours could be parameterized. Thus, we might consider the various 2D flow visualization methods as part of a family of related methods—different kinds of flow oriented contours. Considered in this way, the display problem becomes one of optimizing the various parameters to reveal important aspects of the data for a particular set of tasks and not so much a problem of developing new algorithms.

# Perception of Transparency: Overlapping Data

In many visualization problems, it is desirable to present data in a layered form. This is especially common in geographic information systems (GISs). Sometimes, a useful technique is to present one layer of data as if it were a transparent layer over another. However, there are many perceptual pitfalls in doing this. The contents of the different layers will always interfere with each other to some extent, and sometimes the two layers will fuse perceptually so that it is not possible to determine to which layer a given object belongs.

In simple displays, as in Figure 6.26(a), the two main determinants of perceived transparency are good continuity (Beck and Ivry, 1988) and the ratio of colors or gray values in the different pattern elements. A reasonably robust rule for transparency to be perceived is $x < y < z$ or $x > y > z$ or $y < z < w$ or $y > z > w$, where $x$, $y$, $z$, and $w$ refer to gray values arranged in the pattern shown in Figure 6.26(b) (Masin, 1997). Readers who are interested in perceptual rules of transparency should consult Metelli (1974).

Another way to represent layers of data is to show each layer as a see-through texture or screen pattern (Figure 6.27). Watanabe and Cavanaugh (1996) explored the conditions under which people perceive two distinct overlapping layers, as opposed to a single fused composite texture. They called the effect *laciness*. In Figure 6.27(a) and (b), two different overlapping rectangles are clearly seen, but in (c), only a single textured patch is perceived. In (d), the percept is bistable. Sometimes it looks like two overlapping squares containing patterns of "–" elements; sometimes a central square containing a pattern of "+" elements seems to stand out as a distinct region.

In general, when we present layered data, we can expect the basic rules of perceptual interference, discussed in Chapter 5, to apply. Similar patterns interfere with one another. Graphical patterns that are similar in terms of color, spatial frequency, motion, and so on, tend to interfere more with one another than do those with dissimilar components.

One possible application of transparency in user interfaces is to make pop-up menus transparent so that they do not interfere with information located behind them. Harrison and Vincente (1996) investigated the interference between background patterns and foreground trans-

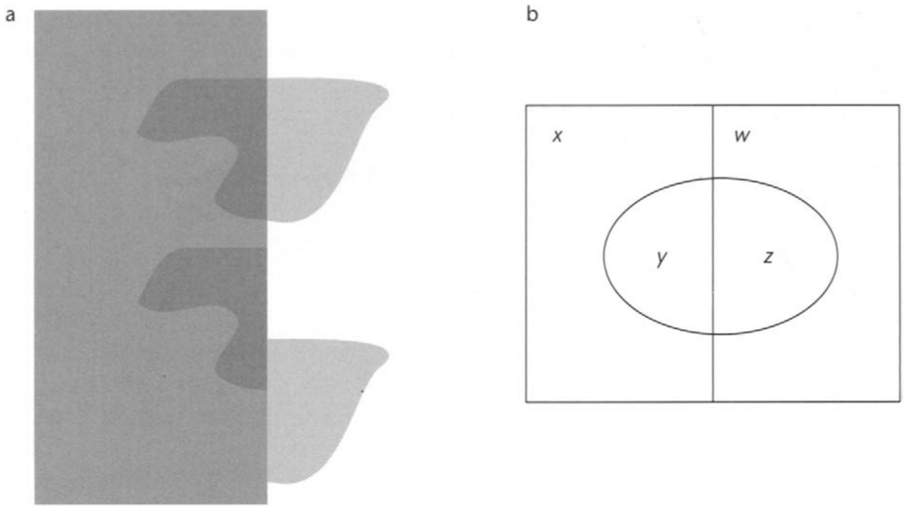



**Figure 6.26** In (a), transparency is perceived only when good continuity is present and when the correct relationship of the colors is present. See text for an explanation of (b).

parent menus. They found that it took longer to read from the menu with text or wireframe drawings in the background than with continuously shaded images in the background. This is exactly what would be expected from an interference model. Because a continuously shaded image lacks the high-frequency detail of a wireframe image or text, there will be less interference between the two. The advantages of transparent layered displays must be weighed against the perceptual interference between the layers. For the designer to minimize visual interference, layers must be maximally separated in the different visual channels. Color, texture, motion, and stereoscopic depth channels can all be used in any combination, depending on the design requirements. The more channels used, the better the separation will be.

# Pattern Learning

If pattern perception is, as claimed, fundamental to extraction of meaning from visualizations, then an important question arises. Can we learn to see patterns better? Artists talk about seeing things that the rest of us cannot see, and ace detectives presumably spot visual clues that are invisible to the beat officer.

What is the scientific evidence that people can learn to see patterns better? The results are mixed. There have been some studies of pattern learning where almost no learning occurred. An often-cited example is the visual search for the simple conjunction of features such as color and shape (Treisman and Gelade, 1980). But other studies have found learning for certain patterns (Logan, 1994). A plausible explanation is that pattern learning occurs least for simple, basic

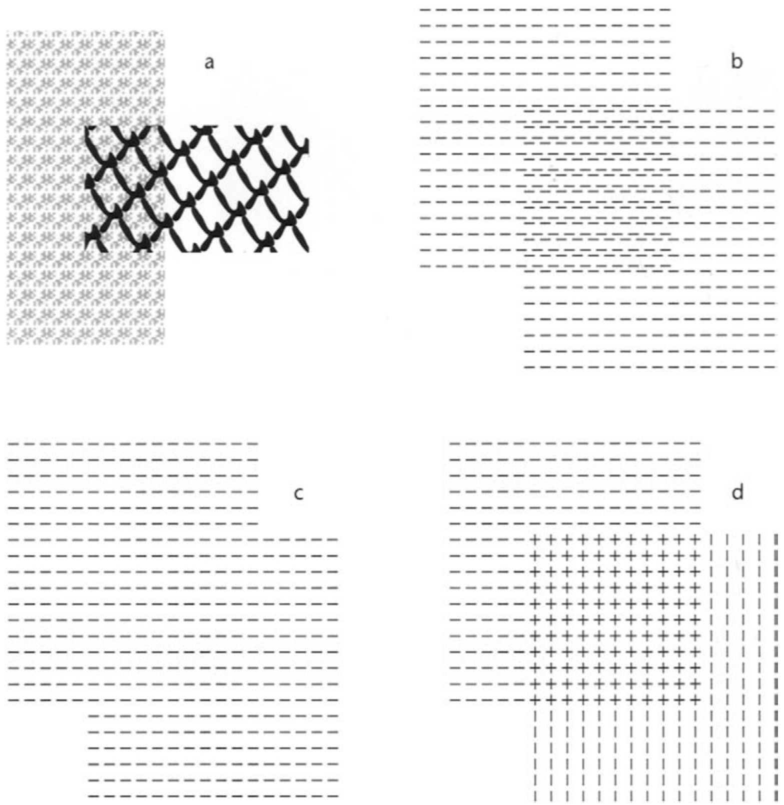



**Figure 6.27** Watanabe and Cavanaugh (1996) called the texture equivalent of transparency *laciness*. This figure is based on their work.

patterns processed early in the visual system, and most for complex, unfamiliar patterns processed late in the visual system.

Fine and Jacobs (2002) reviewed 16 different pattern-learning experiments and found very different amounts of learning. The studies they looked at all contained large numbers of trials (in which a subject would attempt to see a particular pattern in a display) distributed over several days. They found that for simple pattern perception tasks, such as the ability to resolve a grating pattern like that shown in Figure 6.28(a), almost no learning occurred. This task depends on early-stage visual processing, for which the neural machinery is consolidated in the first few months of life. In tasks involving patterns of intermediate complexity, some learning does occur. For example, seeing spatial frequency differences within a pattern such as that shown in Figure 6.28(b) can be learned (Fine and Jacobs, 2000). This is a "plaid" pattern constructed by summing a variety of the sinusoidal gratings. Processing of such patterns is thought to occur

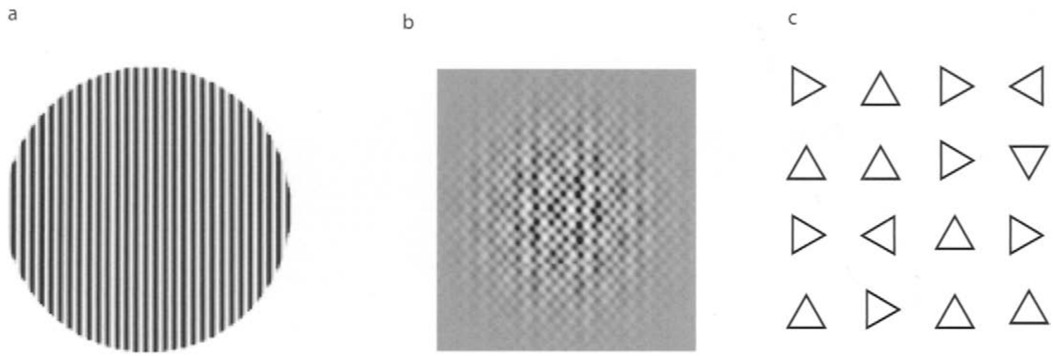



**Figure 6.28** Three patterns used in perceptual learning studies.

mostly at an intermediate stage of the visual system. The most learning was found in higher-level pattern tasks, such as detecting the downward pointing triangles in Figure 6.28(c) (Sigman and Gilbert, 2000).

Another factor that affects learning is the degree to which a particular pattern is already familiar. We would not expect much change in a subject's ability to identify letters of the alphabet in a short experiment, because most people have already been exposed to millions of alphabetic characters. Rapid learning can only be expected for patterns that are unfamiliar. The change in rate of learning over time is captured by the *power law of practice*, which has the following form:

$$\log(T_n) = C - \alpha \log(n) \tag{6.1}$$

This law states that log of the time $T_n$ to respond on the $n^{th}$ trial is inversely proportional to the log of the number of trials. The constant C is the time taken on the first trial (or block of trials).

The power law of practice is usually applied to manual skill learning, but it has also been shown to apply to the perception of complex patterns. Kolers (1975) found that a power law applied to the task of learning to read inverted text. His results are illustrated in Figure 6.29. Initially, it took subjects about 15 minutes to read a single inverted page, but when over 100 pages had been read, the time was reduced to 2 minutes. Although Figure 6.29 shows a straight-line relationship between practice and learning, this is only because of the logarithmic transformation of the data. The relationship is actually very nonlinear. Consider a hypothetical task where people improve by 30% from the first day's practice to the second day. Doubling the amount of practice has resulted in a 30% gain. According to the power law, someone with 10 years of experience at the same task will take a further 10 years to improve by 30%. In other words, practice yields decreasing gains over time.

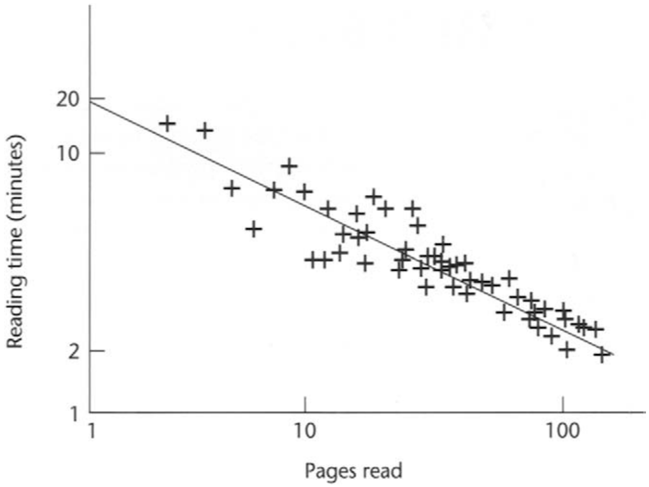



**Figure 6.29** The time to read a page of inverted text is plotted against the number of pages read. Both axes have logarithmic spacing. Data replotted from Newell and Rosenbloom (1981).

In addition to long-term pattern-learning skills, there are also priming effects that are much more transient. Whether these constitute learning is still the subject of debate. *Priming* refers to the phenomenon that once a particular pattern has been recognized, it will be much easier to identify in the next few minutes or even hours. This is usually thought of as a kind of heightened receptivity within the visual system, but some theorists consider it to be visual learning. In either case, once a neural pathway has been activated, its future activation becomes facilitated. For a modern theory of perceptual priming based on neural mechanisms, see Huber and O'Reilly (2003).

What are the implications of these findings for visualization? One is that people can learn pattern-detection skills, although the ease of gaining these skills will depend on the specific nature of the patterns involved. Experts do indeed have special expertise. The radiologist interpreting an X-ray, the meteorologist interpreting radar, and the statistician interpreting a scatter plot will each bring a differently tuned visual system to bear on his or her particular problem. People who work with visualizations must learn the skill of seeing patterns in data. In terms of making visualizations that contain easily identified patterns, one strategy is to rely on pattern-finding skills that are common to everyone. These can be based on low-level perceptual capabilities, such as seeing the connections between objects linked by lines. We can also rely on skill transfer. If we know that our users are cartographers, already good at reading terrain contour maps, we can display other information, such as energy fields, in the form of contour maps. The evidence from priming studies suggests that when we want people to see particular patterns, even familiar ones, it is a good idea to show them a few examples ahead of time.

# The Perceptual Syntax of Diagrams

Diagrams are always hybrids of the conventional and the perceptual. Diagrams contain conventional elements, such as abstract labeling codes, that are difficult to learn but formally powerful. They also contain information that is coded according to perceptual rules, such as Gestalt principles. Arbitrary mappings may be useful, as in the case of mathematical notation, but a good diagram takes advantage of basic perceptual mechanisms that have evolved to perceive structure in the environment. By presenting examples, the following sections describe the visual grammar of two different kinds of diagrams: node–link diagrams and the layered maps used in GISs.

## The Grammar of Node–Link Diagrams

For a mathematician, a graph is a structure consisting of nodes and edges (links between the nodes). See Figure 6.30 for examples. There is a specialized academic field called *graph drawing* whose goal is to make graphs that are pleasantly laid out and easy to read. In graph drawing, layout algorithms are optimized according to aesthetic rules, such as the minimization of link crossings, displaying symmetry of structure and minimizing bends in links (Di Battista et al., 1999). Path bendiness and the number of link crossings have both been shown empirically to degrade performance on the task of finding the shortest path between two nodes (Ware et al., 2002). However, for the most part, there has been little attempt either to systematically apply our knowledge of pattern perception to problems in graph drawing or to use empirical

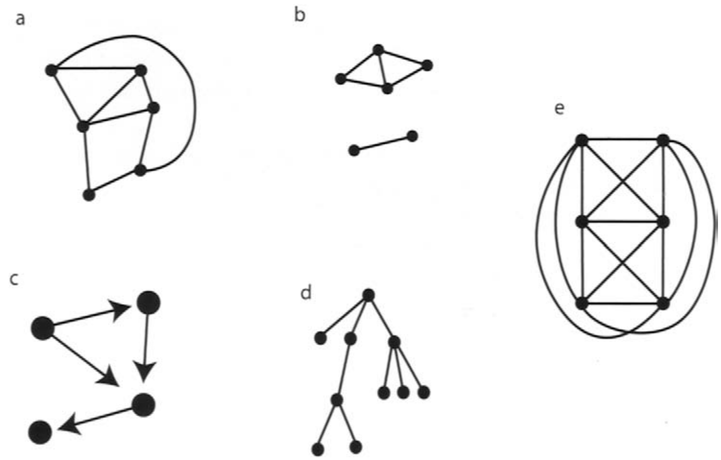



**Figure 6.30** Node–link diagrams, technically called graphs: (a) A graph. (b) A graph with two connected components. (c) A directed graph. (d) A tree structure graph. (e) A nonplanar graph. It cannot be laid out on a plane without links crossing.

methods to determine that graphs laid out according to aesthetic principles are, in fact, easier to understand.

In the following paragraphs, we broaden the concept of a graph to consider a very large class of diagrams that we will call, generically, node–link diagrams. The essential characteristic of these diagrams is that they consist of *nodes*, representing various kinds of entities, and *links*, representing relationships between the entities. Dozens of different diagrams have this basic form, including software structure diagrams, data-flow diagrams, organization charts, and software modeling diagrams. Figure 6.31 provides four examples commonly used in software engineering. The set of abstractions common to node–link diagrams is so close to ubiquitous that it can be called a visual grammar. The nodes are almost always outline boxes or circles, usually representing the entities in a system. The connecting lines generally represent different kinds of relationships, transitions, or communication paths between nodes. Experimental work shows that visualizing *interdependencies* between program elements helps program understanding (Linos et al., 1994).

The various reasons why we may be justified in calling these graphical codes perceptual are distributed throughout this book, but are addressed mostly in this chapter and Chapter 5. The

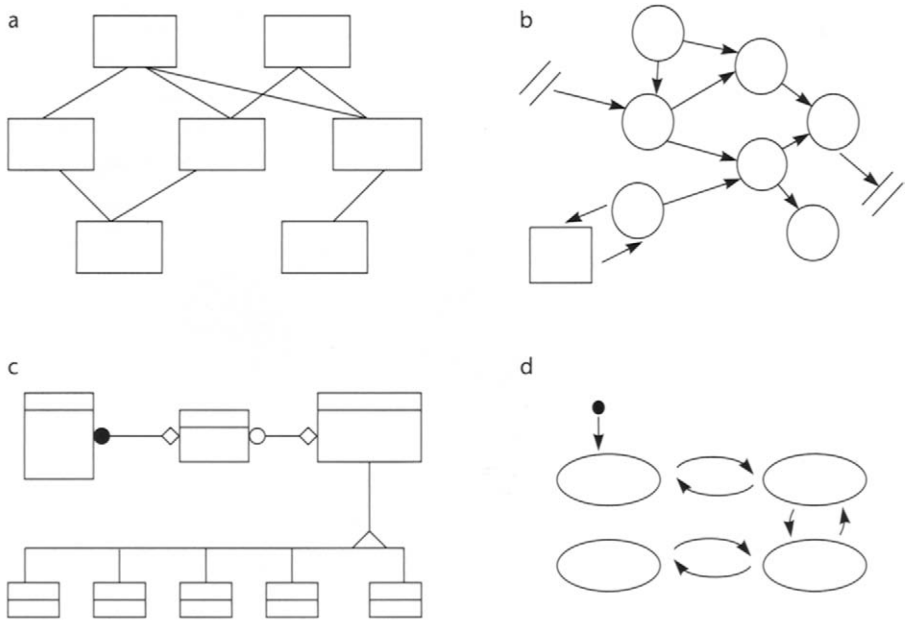



**Figure 6.31** Four different kinds of node–link diagrams used in software engineering: (a) A code module diagram. (b) A data flow diagram. (c) An object modeling diagram. (d) A state transition diagram. Each of these diagrams would normally contain text labels on the nodes and the arcs.

fundamental argument is that closed contours are basic in defining visual objects. Thus, although a circular line may be only a mark drawn on paper, at some level in the visual system it is object-like. Similarly, two objects can be connected by a line, and this visual connection has the ability to represent any of a number of relationships.

Although lines get their expressive power from neural mechanisms designed to interpret objects, they are fundamentally ambiguous. Kennedy (1974) has elucidated many ways in which contours (lines) can represent aspects of the environment. Some of them are illustrated in Figure 6.32. A circle can represent a ring, a flat disk, a ball, a hole, or the boundary between two objects (a disk in a hole). This nicely illustrates the mixture of perception and convention that is common to diagrams. Our visual systems are capable of interpreting a line contour in any of these ways. In real-world scenes, additional information is available to clarify ambiguous contours. In a diagram, the contour may remain perceptually ambiguous and some convention may be necessary to remove the ambiguity. In one kind of diagram, a circle may represent an object; in another, it may represent a hole; in a third, it may represent the boundary of a geographic region. The diagram convention tells us which interpretation is correct.

A general data model that uses a form of node–link diagram is the entity-relationship model. It is widely used in computer science and business modeling (Chen, 1976). In entity relationships, *modeling entities* can be objects and parts of objects, or more abstract things such as parts of organizations. *Relationships* are the various kinds of connections that can exist between entities. For example, an entity representing a wheel will have a part-of relationship to an entity representing an automobile. A person may have a customer relationship to a store. Both entities and relationships can have *attributes*. Thus, a particular customer might be a preferred customer. An attribute of an organization might be the number of its employees. There are standard diagrams for use in entity-relationship modeling, but we are not concerned with these here. We are more

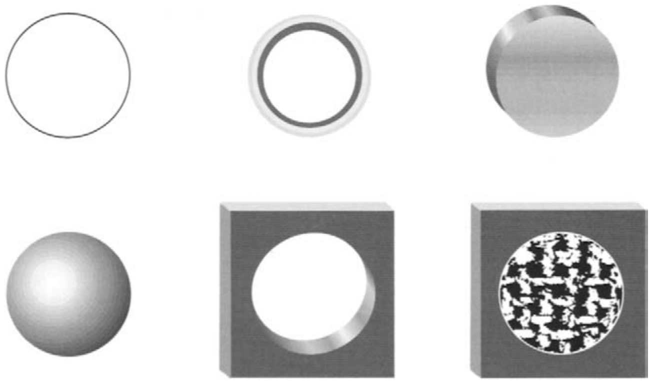


**Figure 6.32** The line circle shown at the top left can represent many kinds of objects: a wire ring, a disk, a ball, a cut-out hole, or the boundary between a disk and the hole in which it resides.

interested in the different ways diagrams can be constructed to represent entities, relationships, and attributes in an easily perceived manner.

The following list is a description of the general ways in which entities and relationships can be expressed using node–link diagrams. This can be regarded loosely as a visual syntax. These are conjectured to be good display mappings, although none has been proved through scientific study to be the best. Each of the elements in the list has a perceptual, rather than conventional, basis for the way it conveys meaning. Most of these elements are discussed more extensively elsewhere in this book. Figure 6.33 provides a set of matching illustrations.

1. A closed contour in a node–link diagram generally represents an entity of some kind. It might be part of a body of software, or a person in an organization.
2. The shape of the closed contour is frequently used to represent an entity type (an attribute of the entity).
3. The color of an enclosed region represents an entity type (an attribute).
4. The size of an enclosed region can be used to represent the magnitude of an entity (a scalar attribute).
5. Lines that partition a region within a closed contour can delineate subparts of an entity. This may correspond to a real-world multipart object.
6. Closed-contour regions may be aggregated. The result is readily seen as a composite entity.
7. A number of closed-contour regions within a larger closed contour can represent conceptual containment.
8. Placing closed contours spatially in an ordered sequence can represent conceptual ordering of some kind.
9. A line linking entities represents some kind of relationship between them.
10. A line linking closed contours can have different colors, or other graphical qualities such as waviness. This effectively represents an attribute or type of a relationship.
11. The thickness of a connecting line can be used to represent the magnitude of a relationship (a scalar attribute).
12. A contour can be shaped with tabs and sockets to indicate which components have particular relationships.
13. Proximity of components can represent groups.

The vast majority of node–link diagrams currently in use are very simple. For the most part, these diagrams use identical rectangular or circular nodes and constant-width lines, like those shown in Figure 6.31. Although such generic diagrams are very effective in conveying patterns of structural relationships among entities, they are often poor at showing the types of entities

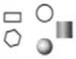
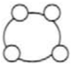
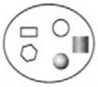
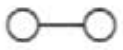
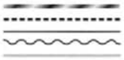
| Graphical Code | Visual Instantiation | Semantics |
|---|---|---|
| 1. Closed contour. |  | Entity, object, node. |
| 2. Shape of closed region. |  | Entity type. |
| 3. Color of enclosed region. |  | Entity type. |
| 4. Size of enclosed region. |  | Entity value. Larger = more. |
| 5. Partitioning lines within enclosed region. |  | Entity partitions are created, e.g., TreeMaps. |
| 6. Attached shapes. |  | Attached entities. Part-of relations. |
| 7. Shapes enclosed by contour. |  | Contained entities. |
| 8. Spatially ordered shapes. |  | A sequence. |
| 9. Linking line. |  | Relationship between entities. |
| 10. Linking-line quality. |  | Type of relationship between entities. |
| 11. Linking-line thickness. |  | Strength of relationship between entities. |
| 12. Tab connector. |  | A fit between components. |
| 13. Proximity. |  | Groups of components. |

**Figure 6.33** The visual grammar of diagram elements (node–link diagrams).

and the types of relationships. Attributes, when they are shown, are often provided in the form of text labels attached to the boxes and lines, although occasionally dashed lines and other variations are used to denote types.

Clearly, there are ways to extend this vocabulary that are perceptually sound. Chapter 7 introduces the concept of a *geon diagram* as a graphical device that uses 3D objects, with surface texture and color, to represent entities and relationships. There is a range of possibilities between the rectangular-box-and-line diagram and fully rendered, colored, and textured 3D objects. We can make diagram boxes that are more objectlike, with shape and texture denoting various attributes, and we can depict relationships using thin tubes. Most of the different ways of representing attributes shown in Figure 6.33 are rarely used, although they are relatively easy to implement with modern computer graphics.

## The Grammar of Maps

A second visual grammar can be found in the way maps are designed and interpreted. Only three basic kinds of graphical marks are common to most maps: areas, line features, and point features (Mark and Franck, 1996). Figure 6.34 illustrates this basic grammar of maps and shows how these three elements can work in isolation and in combination.

1, 2, 3. Geographical areas are usually denoted by closed contours, tinted areas, or textured areas. Often, in a map, all three methods can be used; for example, lines to represent county boundaries, color-coding to represent climate, and texture to represent vegetation.

4. Geographical linear features represent either boundaries or elongated geographical regions. The difference between geographical areas and linear features is sometimes related to scale. At a small scale, a river will be represented by a thin line of constant width; at a larger scale, it can become an extended geographical area.
5. Dots or other small symbols are used to represent *point features*, although whether or not something is a point feature depends on the scale. At a large scale, an entire city may be represented by a single dot; at a small scale, a dot might be used to show the locations of churches, schools, or tourist attractions.
6. A dot on a line means that the entity denoted by the point feature is on, or attached to, the entity denoted by the linear feature. For example, a city is "on" a river.
7. A dot within a closed contour means that the entity denoted by the point feature lies within the boundaries of the area feature. For example, a town is within a province.
8. A line crossing a closed-contour region means that a linear feature traverses an area feature. For example, a road passes through a county.
9. A line that ends in a closed-contour region means that a linear feature ends or starts within an area feature. For example, a river flows out of a park.
10. Overlapping contour regions denoted by contour, color, or texture denote overlapping spatial entities. For example, a forested region may overlap a county boundary.

| Graphical Code | Visual Instantiation | Semantics |
|---|---|---|
| 1. Closed contour. |  | Geographic region. |
| 2. Colored region. |  | Geographic region. |
| 3. Textured region. |  | Geographic region. |
| 4. Line. |  | Linear map features such as rivers, roads, etc. Depends on scale. |
| 5. Dot. |  | Point features such as town, building. Depends on scale. |
| 6. Dot on line. |  | Point feature such as town on linear feature such as road. |
| 7. Dot in closed contour. |  | Point feature such as town located within a geographic region. |
| 8. Line crosses closed-contour region. |  | Linear feature such as river crossing geographic region. |
| 9. Line exits closed-contour region. |  | A linear feature such as a river terminates in a geographic region. |
| 10. Overlapping contour, colored regions, textured regions. |  | Overlapping geographically defined areas. |

**Figure 6.34** The visual grammar of map elements.

Maps need not be used only for geographical information. Johnson and Shneiderman (1991) developed a visualization technique they call a *treemap*, for displaying information about the tree data structures commonly used in computer science. Figure 6.35 shows an example of a tree data structure presented in treemap form and in a conventional node–link diagram.

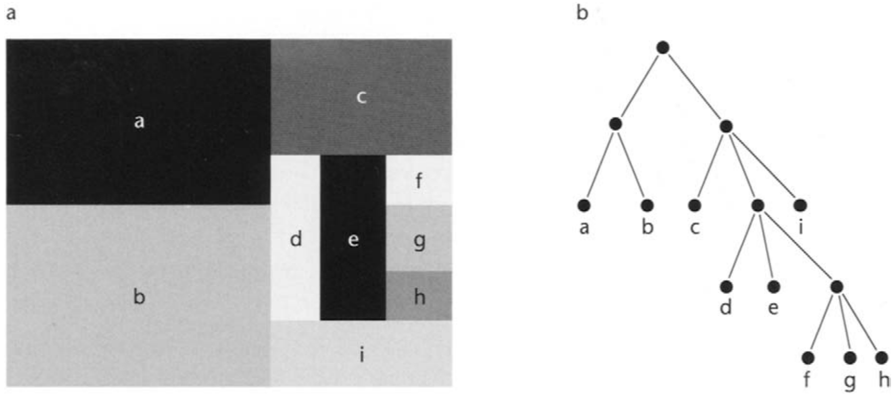



**Figure 6.35** (a) A treemap representation of hierarchical data. Areas represent the amount of data stored in parts of the tree data structure. (b) The same tree structure, represented using a node–link diagram.

The original treemap was based on the following algorithm. First the rectangle is divided with a vertical partition according to the number of branches from the root of the tree. Next, each subrectangle is similarly divided, but with horizontal partitions. This process is repeated to the "leaves" of the tree. The area of each leaf on the tree corresponds to the amount of information that is stored there.

The great advantage of the treemap over conventional tree views is that the amount of information on each branch of the tree can be easily visualized. Because the method is space-filling, it can show quite large trees containing thousands of branches. The disadvantage is that the hierarchical structure is not as clear as it is in a more conventional tree drawing, which is a specialized form of node–link diagram.

# Patterns in Motion

To this point, we have mainly discussed the use of static patterns to represent data, even though the data is sometimes dynamic—as in the case of a vector field representing a pattern of moving liquid or moving gas. We can also use motion as a display technique to represent data that is either static or dynamic. The perception of dynamic patterns is not understood as well as the perception of static patterns. But we are very sensitive to patterns in motion and, if we can learn to use motion effectively, it may be a good way to display certain aspects of data.

We start by considering the problem of how to represent data communications with computer animation. One way of doing this is to use a graphical object to represent each packet of information and then to animate that package from the information source to its destination.

First we consider the simplest case—data represented by a series of identical and equally spaced graphical elements, as shown in Figure 6.36. In this case, there is a fundamental limitation on the throughput that can be represented. In a computer animation sequence, the basic process is a loop that involves drawing the animated object, displaying it, moving it, and then redrawing it. When this cycle is repeated fast enough, a sequence of static pictures is seen as a smoothly moving image. The limitation on perceived data throughput arises from the amount that a given object can be moved before it becomes confused with another object in the next frame—this is called the *correspondence problem.*

If we define the distance between pattern elements as $\lambda$, we are limited to a maximum displacement of $\lambda/2$ on each frame of animation before the pattern is more likely to be seen as moving in the reverse direction from that desired. The problem is illustrated in Figure 6.36(a).

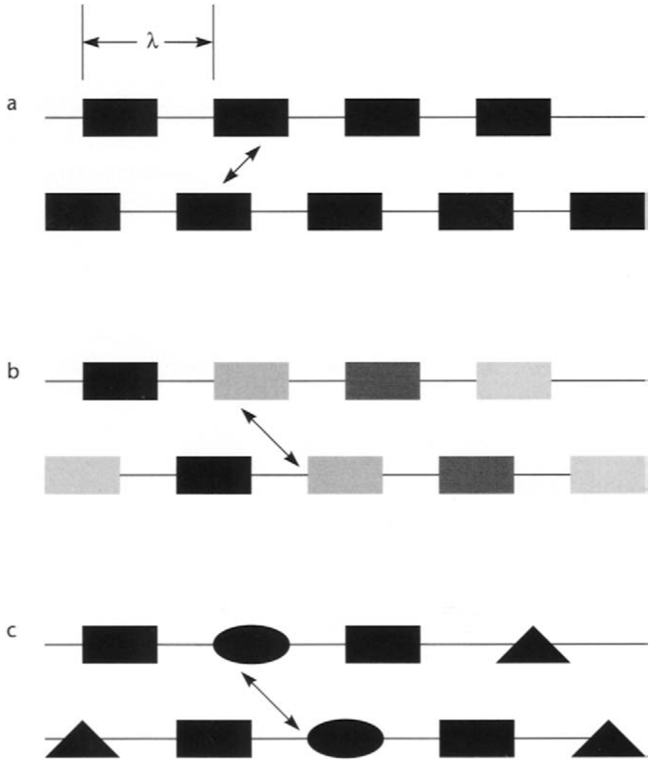



**Figure 6.36** If motion is represented using a regular sequence of identical and equally spaced elements, there is a strict limit on the throughput that can be perceived. This limit can be extended by varying the sizes and shapes of the graphical elements.

When all the elements are identical, the brain constructs correspondences based on object proximity in successive frames. This is sometimes called the *wagon-wheel effect*, because of the tendency of wagon wheels in Western movies to appear to be rotating in the wrong direction. Experiments by Fleet (1998) suggest that the maximum change per frame of animation for motion to reliably be seen in a particular direction is about $\lambda/3$ for the basic representation shown in Figure 6.36(a). Given an animation frame rate of 60 frames per second, this establishes an upper bound of 20 messages per second that can be represented.

There are many ways in which the correspondence limitation can be overcome by giving the graphical elements a different shape, orientation, or color. Two possibilities are illustrated in Figure 6.36(b) and (c). In one, the gray values of the elements are varied from message to message; in the other, the shapes of the elements are varied. Research with element shapes suggests that correspondence of shape is more important than correspondence of color in determining perceived motion (Caelli et al., 1993). In a series of experiments that examined a variety of enhanced representations like those illustrated in Figure 6.36(b) and (c), Fleet (1998) found that the average phase shift per animation frame could be increased to $3\lambda$ before correspondence was lost. Given an animation frame rate of 60 frames per second, this translates to an upper bound of 180 messages per second that can be represented using animation.

Of course, when the goal is to visualize high traffic rates, there is no point in representing individual messages in detail. Most digital communications systems transfer millions of data packets per second. What is important at high data rates is an impression of data volumes, the direction of traffic flow, and large-scale patterns of activity.

## Form and Contour in Motion

A number of studies have shown that people can see relative motion with great sensitivity. For example, contours and region boundaries can be perceived with precision in fields of random dots if defined by differential motion alone (Regan, 1989; Regan and Hamstra, 1991). Human sensitivity to such motion patterns rivals our sensitivity to static patterns; this suggests that motion is an underutilized method for displaying patterns in data.

For purposes of data display, we can treat motion as an attribute of a visual object, much as we consider size, color, and position to be object attributes. We evaluated the use of simple sinusoidal motion in enabling people to perceive correlations between variables (Limoges et al., 1989). We enhanced a conventional scatter plot representation by allowing the points to oscillate sinusoidally, either horizontally or vertically (or both) about a center point. An experiment was conducted to discover whether the frequency, phase, or amplitude of point motion was the most easily "read." The task was to distinguish a high correlation between variables from a low one. A comparison was made with more conventional graphical techniques, including using point size, gray value, and $x,y$ position in a conventional scatter plot. The results showed that data mapped to phase was perceived best; in fact, it was as effective as most of the more conventional techniques, such as the use of point size or gray value. In informal studies, we also showed that motion appears to be effective in revealing clusters of distinct data points in a multidimensional data space (see Figure 6.37). Related data shows up as clouds of points moving together in elliptical paths, and these can be easily differentiated from other clouds of points.

## Moving Frames

Perceived motion is highly dependent on its context. Johansson (1975) has demonstrated a number of grouping phenomena that show that the brain has a strong tendency to group moving objects in a hierarchical fashion. One of the effects he investigated is illustrated in Figure 6.38. In this example, three dots are set in motion. The two outer dots move in synchrony in a horizontal direction. The third dot, located between the other two, also moves in synchrony but in an oblique direction. However, the central dot is not perceived as moving along an oblique path. Instead, what is perceived is illustrated in 6.38(b). An overall horizontal motion of the entire group of dots is seen; within this group, the central dot also appears to move vertically.

A rectangular frame provides a very strong contextual cue for motion perception. It is so strong that if a bright frame is made to move around a bright static dot in an otherwise

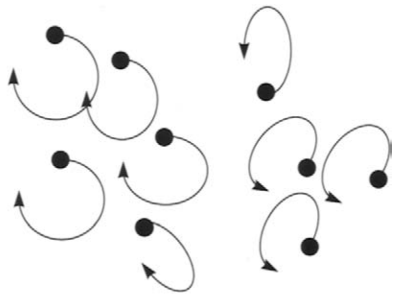


**Figure 6.37** An illustration of the elliptical motion paths that result when variables are mapped to the relative phase angles of oscillating dots. The result is similar elliptical motion paths for points that are similar. In this example, two distinct groups of oscillating dots are clearly perceived.

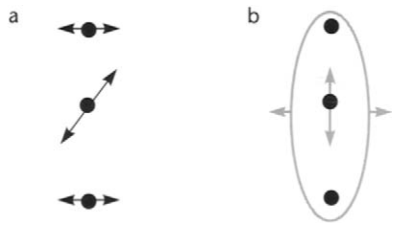



**Figure 6.38** When dots are set in synchronized motion, as shown in (a), what is actually perceived is shown in (b). The entire group of dots is seen to move horizontally, and the central dot moves vertically within the group.

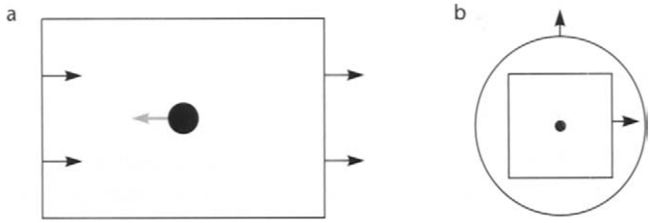



**Figure 6.39** (a) When a stationary dot is placed within a moving frame in a dark room, it is the dot that is perceived to move in the absence of other cues. (b) The effect is hierarchical.

completely dark environment, it is often the static dot that appears to move (Wallach, 1959). Wallach also showed that the effect works in a hierarchical fashion. Thus, the perceived motion of the static dot in Figure 6.39(b) is strongly influenced by the motion of a surrounding square frame, but it is much less influenced by the motion of the circle outside the square.

Computer animation is often used in a straightforward way to display dynamic phenomena, such as a particle flow through a vector field. In these applications, the main goal from a perceptual point of view is to bring the motion into the range of human sensitivities. The issue is the same for viewing high-speed or single-frame movie photography. The motions of flowers blooming or bullets passing through objects are speeded up and slowed down, respectively, so that we can perceive the dynamics of the phenomena. Humans are reasonably sensitive to motion ranging from a few millimeters per second to a few hundred millimeters per second for objects viewed at normal screen distances. Generally, the data animator should aim for motion in the midrange of a few centimeters per second. (See Chapter 2 for some of the basic issues related to motion sensitivity.)

The use of motion to help us distinguish patterns in abstract data is at present only a research topic, albeit a very promising one. One application of the research results is the use of frames to examine dynamic flow field animations. Frames can be used as an effective device for highlighting local relative motion. If we wish to highlight the local relative motion of a group of particles moving through a fluid, a rectangular frame that moves along with the group will create a reference area within which local motion patterns can emerge.

Another way in which motion patterns are important is in helping us to perceive visual space and rigid 3D shapes. This topic is covered in Chapter 8 in the context of the other mechanisms of space perception.

## Expressive Motion

Using moving patterns to represent motion on communication channels, or in vector fields, is a rather obvious use of motion for information display, but there are other, more subtle uses. There appears to be a vocabulary of expressive motion comparable in richness and variety to the vocabulary of static patterns explored by the Gestalt psychologists. In the following sections, some of the more provocative results are discussed, together with their implications for data visualization.

## Perception of Causality

When we see a billiard ball strike another and set the second ball in motion, we perceive that the motion of the first ball *causes* the motion of the second, according to the work of Michotte (translated 1963). Michotte's book *The Perception of Causality* is a compendium of dozens of experiments, each showing how variations in the basic parameters of velocity and event timing can radically alter what is perceived. He conducted detailed studies of the perception of interactions between two patches of light and came to the conclusion that the perception of causality can be as direct and immediate as the perception of simple form. In a typical experiment, illustrated in Figure 6.40, one rectangular patch of light moved from left to right until it just touched a second patch of light, then stopped. At this point, the second patch of light would start to move. This was before the advent of computer graphics, and Michotte conducted his experiments with an apparatus that used little mirrors and beams of light. Depending on the temporal relationships between the moving-light events and their relative velocities, observers reported different kinds of causal relationships, variously described as "launching," "entraining," or "triggering."

Precise timing is required to achieve perceived causality. For example, Michotte found that for the effect he called *launching* to be perceived, the second object had to move within 70 milliseconds of contact; after this interval, subjects still perceived the first object as setting the second object in motion, but the phenomenon was qualitatively different. He called it *delayed launching*. Beyond about 160 milliseconds, there was no longer an impression that one event caused the other; instead, unconnected movements of the two objects were perceived. Figure 6.41 provides a reproduction of some of his results. For causality to be perceived, visual events must be synchronized within at least one-sixth of a second. Given that virtual-reality animation often occurs at only about 10 frames per second, events should be frame-accurate for clear causality to be perceived.

If an object makes contact with another and the second object moves off at a much greater velocity, a phenomenon that Michotte called *triggering* is perceived. The first object does not seem to cause the second object to move by imparting its own energy; rather, it appears that contact triggers propelled motion in the second object.

**Figure 6.40** Michotte (1963) studied the perception of causal relationships between two patches of light that always moved along the same line but with a variety of velocity patterns.

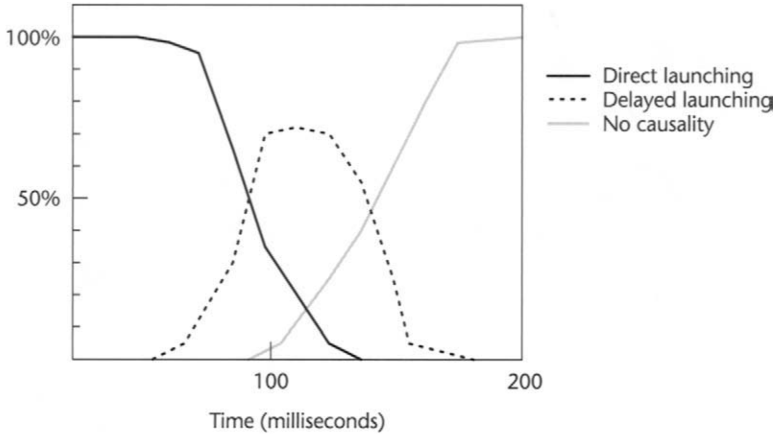



**Figure 6.41** From Michotte (1963). When one object comes into contact with another and the second moves off, the first motion may be seen to cause the second if the right temporal relationships exist. The graph shows how different kinds of phenomena are perceived, depending on the delay between the arrival of one object and the departure of the other.

More recent developmental work by Leslie and Keeble (1987) has shown that infants at only 27 weeks of age can perceive causal relations such as launching. This would appear to support the contention that such percepts are in some sense basic to perception.

The significance of Michotte's work for data visualization is that it provides a way to increase the expressive range beyond what is possible with static diagrams. In a static visualization, the visual vocabulary for representing relationships is quite limited. To show that one visual object is related to another, we can draw lines between them, we can color or texture groups of objects, or we can use some kind of simple shape coding. The only way to show a causal link between two objects is by using some kind of conventional code, such as a labeled arrow. However, such codes owe their meaning more to our ability to understand conventional coded language symbols than to anything essentially perceptual. This point about the differences between language-based and perceptual codes is elaborated in Chapter 9. What Michotte's work gives us is the ability to significantly enrich the vocabulary of things that can be immediately and directly represented in a diagram.

## Perception of Animate Motion

In addition to the fact that we can perceive causality using simple animation, there is evidence that we are highly sensitive to motion that has a biological origin. In a series of now-classic studies, Gunnar Johansson attached lights to the limb joints of actors (Johansson, 1973). He then produced moving pictures of the actors carrying out certain activities, such as walking and

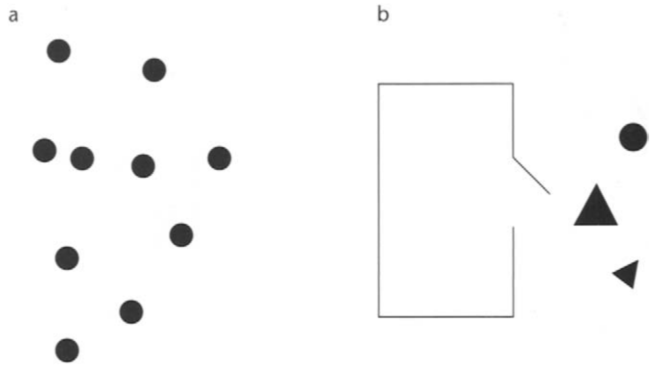



**Figure 6.42** (a) In Johansson's (1973) experiments, a pattern of moving dots was produced by making a movie of actors with lights attached to parts of their bodies. (b) Heider and Semmel (1944) made a movie of simple geometric shapes moving through complex paths. Viewers of both kinds of displays attribute anthropomorphic characteristics to what they see.

dancing. These pictures were made so that only the points of light were visible, and, in any given still frame, all that was perceived was a rather random-looking collection of dots, as shown in Figure 6.42(a). A remarkable result from Johansson's studies was that viewers of the animated movies were immediately conscious of the fact that they were watching human motion. In addition, they could identify the genders of the actors and the tasks they were performing. Some of these identifications could be made after exposures lasting only a small fraction of a second.

Another experiment pointing to our ability to recognize form from motion was a study by Heider and Semmel (1944). In this study, an animated movie was produced incorporating the motion of two triangles and a circle, as shown in Figure 6.42(b). People viewing this movie readily attributed human characteristics to the shapes; they would say, for example, that a particular shape was angry, or that the shapes were chasing one another. Moreover, these interpretations were consistent across observers. Because the figures were simple shapes, the implication is that patterns of motion were conveying the meaning. Other studies support this interpretation. Rimé et al. (1985) did a cross-cultural evaluation of simple animations using European, American, and African subjects, and found that motion could express such concepts as kindness, fear, or aggression, and there was considerable similarity in these interpretations across cultures, suggesting some measure of universality.

## Enriching Diagrams with Simple Animation

The research findings of Michotte, Johansson, Rimé, and others suggest that the use of simple motion can powerfully express certain kinds of relationships in data. Animation of abstract shapes can significantly extend the vocabulary of things that can be conveyed naturally beyond what is possible with a static diagram. The key result, that motion does not require the support of complex depictive representations (of animals or people) to be perceived as animate, means that simplified motion techniques may be useful in multimedia presentations. The kinds of animated critters that are starting to crawl and hop over Web pages are often unnecessary and distracting. Just as elegance is a virtue in static diagrams, so is it a virtue in diagrams that use animation. A vocabulary of simple expressive animation requires development, but research results strongly suggest that this will be a productive and worthwhile endeavor. The issue is pressing, because animation tools are becoming more widely available for information display systems. More design work and more research are needed.

# Conclusion

The brain is a powerful pattern-finding engine; indeed, this is the fundamental reason why visualization techniques are becoming important. There is no other way of presenting information so that structures, groups, and trends can be discovered among hundreds of data values. If we can transform data into the appropriate visual representation, its structure may be revealed. However, not all patterns are equally easy to perceive. The brain appears to be especially good at discovering linear features and distinct objects, so much so that the discovery of spurious patterns should always be a concern. Because the brain is a pattern-finding engine, patterns may be perceived even where there is only visual noise.

Much of the material presented in this chapter, especially the Gestalt laws of pattern perception, leads to rules that seem obvious to any visual designer. Nevertheless, it is surprising how often these design rules are violated. A common mistake is that related data glyphs are placed far apart in displays. Another is that closed contours are used in ways that visually segment a display into regions that make it difficult, rather than easy, to comprehend related information. The use of windows is often to blame, because they result in strong framing effects, which can cause confusion if used inconsistently.

For information to be clearly related, the visual structure should reflect relationships between data entities. Placing data glyphs in spatial proximity, linking them with lines, or enclosing them within a contour will provide the necessary visual structure to make them seem related. In terms of seeing patterns in rather abstract data displays, perception of contours is likely to be especially important. The visual system contains a number of mechanisms for finding contours. These contours can be simple lines, dots, or other features in a linear pattern; boundaries between regions of different textures, different colors, different motion; or even illusory contours.

For the researcher and for those interested in finding novel display techniques, the effective use of motion is suggested as a fertile area for investigation. Patterns in moving data points can be perceived easily and rapidly. Given the computing power of modern personal computers, the opportunity exists to make far greater use of animation in visualizing information.

In considering pattern perception, we should always bear in mind that the perception of abstract patterns is probably not a primary purpose of visual perception. Rather, pattern-finding mechanisms are part of the neural machinery that divides the world into visual objects. For example, the reason that closed contours are so compelling in segmenting space is that they normally define objects in our environment; they do not have any special significance in and of themselves. In the next chapter, we consider ways in which 3D objects are perceived and ways in which object displays can be used to organize information.