CHAPTER 10

Interacting with Visualizations

A good visualization is not just a static picture or a 3D virtual environment that we can walk through and inspect like a museum full of statues. A good visualization is something that allows us to drill down and find more data about anything that seems important. Ben Shneiderman has coined what he calls a "mantra" to guide visual information-seeking behavior and the interfaces that support it: "Overview first, zoom and filter, then details on demand," (Shneiderman, 1998). But in reality we are just as likely to see an interesting detail, zoom out to get an overview, find some related information in a lateral segue, and then zoom in again to get the details of the original object of interest. The important point is that a good computer-based visualization is an interface that can support all of these activities. Ideally, every data object on a screen will be active and not just a blob of color on the screen. It will be capable of displaying more information as needed, disappearing when not needed, and accepting user commands to help with the thinking processes.

Interactive visualization is a process made up of a number of interlocking feedback loops that fall into three broad classes. At the lowest level is the *data manipulation loop,* through which objects are selected and moved using the basic skills of eye–hand coordination. Delays of even a fraction of a second in this interaction cycle can seriously disrupt the performance of higher-level tasks. At an intermediate level is an *exploration and navigation loop,* through which an analyst finds his or her way in a large visual data space. As people explore a new town, they build a cognitive spatial model using key landmarks and paths between them; something similar occurs when they explore data spaces.

But exploration can be generalized to more abstract searching operations. Kirsh and Maglio (1994) define a class of epistemic actions as activities whereby someone hopes to better understand or perceive a problem. At the highest level is a *problem-solving loop* through which the analyst forms hypotheses about the data and refines them through an augmented visualization process. The process may be repeated through multiple visualization cycles as new data is added, the problem is reformulated, possible solutions are identified, and the visualization is revised or

replaced. Sometimes the visualization may act as a critical externalization of the problem, forming a crucial extension of the cognitive process.

This chapter deals with two of the three loops: low-level interaction and exploration. General problem solving is discussed in Chapter 11.

Data Selection and Manipulation Loop

There are a number of well established "laws" that describe the simple, low-level control loops needed in tasks such as the visual control of hand position or the selection of an object on the screen.

Choice Reaction Time

Given an optimal state of readiness, with a finger poised over a button, a person can react to a simple visual signal in about 130 msec (Kohlberg, 1971). If the signals are very infrequent, the time can be considerably longer. Warrick et al. (1964) found reaction times as long as 700 msec under conditions such that there could be as much as two days between signals. The participants were engaged in routine typing, so they were at least positioned appropriately to respond. If people are not positioned at workstations, their responses will naturally take longer.

Sometimes, before someone can react to a signal, he or she must make a choice. A simple choice reaction-time task might involve pressing one button if a red light goes on and another if a green light goes on. This kind of task has been studied extensively. It has been discovered that reaction times can be modeled by a simple rule called the *Hick–Hyman law* for choice reaction time (Hyman, 1953).

According to this law,

$$\text{Reaction time} = a + b \log_2(C) \tag{10.1}$$

where C is the number of choices and a and b are empirically determined constants. The expression $\log_2(C)$ represents the amount of information processed by the human operator, expressed in bits of information.

Many factors have been found to affect choice reaction time—the distinctness of the signal, the amount of visual noise, stimulus–response compatibility, and so on—but under optimal conditions, the response time per bit of information processed is about 160 msec plus the time to set up the response. Thus, if there are eight choices (3 bits of information), the response time will typically be on the order of the simple reaction time plus approximately 480 msec. Another impor-

tant factor is the degree of accuracy required—people respond faster if they are allowed to make mistakes occasionally, and this effect is called a *speed–accuracy tradeoff*. For a useful overview of factors involved in determining reaction time, see Card et al. (1983).

2D Positioning and Selection

In highly interactive visualization applications, it is useful to have graphical objects function not only as program output—a way of representing data—but also as program input, a way of finding out more about data.

Selection using a mouse or some similar input device (such as a joystick or trackball) is one of the most common interactive operations in the modern graphical user interface, and it has been extensively studied. A simple mathematical model provides a useful estimation of the time taken to select a target that has a particular position and size:

$$\text{Selection time} = a + b \log_2(D/W + 1.0) \tag{10.2}$$

where D is the distance to the center of the target, W is the width of the target, and a and b are constants determined empirically. These are different for different devices.

This formula is known as Fitts' law, after Paul Fitts (1954). The term $\log_2 (D/W + 1.0)$ is known as the *index of difficulty* (ID). The value $1/b$ is called the *index of performance* (IP) and is given in units of bits per second. There are a number of variations in the index-of-difficulty expression, but the one given here is the most robust (MacKenzie, 1992). Typical IP values for measured performance made with the fingertip, the wrist, and the forearm are all in the vicinity of 4 bits per second (Balakrishnan and MacKenzie, 1997). To put this into perspective, consider moving a cursor 16 cm across a screen to a small (0.5 cm) target. The index of difficulty will be about 5 bits. The selection will take more than a second longer than selecting a target that is already under the cursor.

Fitts' law can be thought of as describing an iterative process of eye–hand coordination, as illustrated in Figure 10.1. The human starts by judging the distance to the target and initiates the hand movement. On successive iterations, a corrective adjustment is made to the hand movement based on visual feedback showing the cursor position. The number of iterations around the control loop increases both as the distance to the target gets larger and as the size of the target gets smaller. The logarithmic nature of the relationship derives from the fact that on each iteration, the task difficulty is reduced in proportion to the remaining distance.

In many of the more complex data visualization systems, as well as in experimental data visualization systems using 3D virtual-reality (VR) technologies, there is a significant lag between a hand movement and the visual feedback provided on the display (Liang et al., 1991; Ware and Balakrishnan, 1994).

Fitts' law, modified to include lag, looks like this:

$$\text{Mean time} = a + b\,(\text{Human Time} + \text{MachineLag})\log_2(D/W + 1.0) \qquad \textbf{(10.3)}$$

According to this equation, the effects of lag increase as the target gets smaller. Because of this, a fraction-of-a-second lag can result in a subject's taking several seconds longer to perform a simple selection task. This may not seem like much, but in a VR environment intended to make everything seem easy and natural, lag can make the simplest task difficult.

Fitts' law is part of ISO standard 9214-9, which sets out protocols for evaluating user performance and comfort when using pointing devices with visual display terminals. It is invaluable as a tool for evaluating potential new input devices.

Hover Queries

The most common kind of selection action with a computer is done by dragging a cursor over an object and clicking the mouse button. The *hover query* dispenses with the mouse click. Extra information is revealed about an object when the mouse cursor passes over it. Usually it is implemented with a delay; for example, the function of an icon is shown by a brief text message after

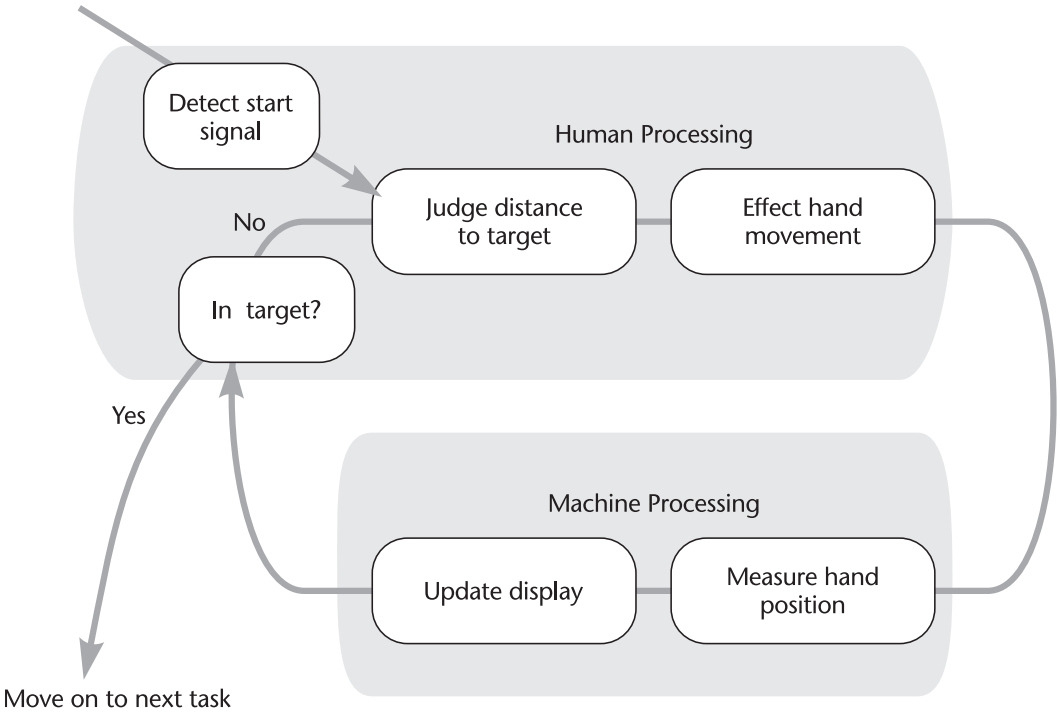

Figure 10.1 The visually guided reaching control loop. The human processor makes adjustments based on visual feedback provided by the computer.

hovering for a second or two. However, a hover query can function without a delay, making it very fast. The enables the mouse cursor to be dragged over a set of data objects, rapidly revealing the data contents and perhaps allowing an interactive query rate of several per second in special circumstances.

Path Tracing

Fitts' law deals with single, discrete actions, such as reaching for an object. Other tasks, such as tracing a curve or steering a car, involve continuous ongoing control. In such tasks, we are continually making a series of corrections based on visual feedback about the results of our recent actions. Accot and Zhai (1997) used Fitts' law to derive a prediction about continuous steering tasks. Their derivation revealed that the speed at which tracing could be done should be a simple function of the width of the path:

$$v = W/\tau \tag{10.4}$$

where v is the velocity, W is the path width, and τ is a constant that depends on the motor control system of the person doing the tracing. In a series of experiments, the researchers found an almost perfect linear relationship between speed of path-following and the path width, confirming their theory. The actual values of τ lay between .05 and .11 sec, depending on the specific task. To make this more concrete, consider the problem of tracing a pencil along a 2 mm–wide path. Their results suggest that this will be done at a rate of between 1.8 and 4 cm/sec.

Two-Handed Interaction

In most computer interfaces, users select and move graphical objects around the screen with a mouse held in one hand, leaving the other unoccupied. But in interacting with the everyday world, we frequently use both our hands. This leads us to the question of how we might make the computer interface better by taking advantage of both hands (Buxton and Myers, 1986).

The most important principle that has been discovered relating to the way tasks should be allocated to the two hands is Guiard's *kinematic chain theory* (Guiard, 1987). According to this theory, the left hand and the right hand form a kinematic chain, with the left hand providing a frame of reference for movements with the right, in right-handed individuals. For example, if we sculpt a small object out of modeling clay, we are likely to hold it in the left hand and do the detailed shaping with the right. The left hand reorients the piece and provides the best view, whereas the right pokes and prods within that frame of reference.

A number of interface designers have incorporated this principle into demonstrably superior interfaces for various tasks (Bier et al., 1993, Kabbash et al., 1994). For example, in an innovative computer-based drawing package, Kurtenbach et al. (1997) showed how templates, such as the French curve, could be moved rapidly over a drawing with the left hand while an artist used his or her right hand to paint around the shape.

Another way that using the left hand can be beneficial is in positioning tools for easy access. In interactive drawing packages, users spend a lot of time moving between the drawing and various menus, positioned off to the side of the screen. The *toolglass* and *magic lens* approach, developed by Bier et al. (1993), got around this problem by allowing users to use the left hand to position tool palettes and the right hand to do normal drawing operations. This allowed for very quick changes in color or brush characteristics. As an additional design refinement, they also made some of the tools transparent (hence toolglasses).

In an application more relevant to information visualization, Stone et al. (1994) developed the magic lens idea as a set of interactive information filters implemented as transparent windows that the user can move over an information visualization with the left hand. The magic lens could be programmed to be a kind of data X-ray, revealing normally invisible aspects of the data. For example, a magic lens view of a map might show some or all of the regions with high rainfall, or alternatively, the geology underneath. In the magic lens design, the right hand can be used in a conventional way, to control a cursor that can then be used to click within the magic lens, to make selections or position objects.

Learning

Over time, people become more skilled at any task, barring fatigue, sickness, or injury. A simple expression known as the *power law of practice* describes the way task performance speeds up over time.

$$\log(T_n) = C - \alpha \log(n) \tag{10.5}$$

where $C = \log(T_1)$ is based on the time to perform the task on the first trial and, T_n is the time required to perform the nth trial, and α is a constant that represents the steepness of the learning curve.

One of the ways in which skilled performance is obtained is through the *chunking* of small subtasks into programmed motor procedures. The beginning typist must make a conscious effort to hit the letters *t, h,* and *e* when typing the word *the,* but the brains of experienced typists can execute preprogrammed bursts of motor commands so that the entire word can be typed with a single mental command to the motor cortex. Skill learning is characterized by more and more of the task becoming automated and encapsulated. To encourage skill automation, the computer system should provide rapid and clear feedback of the consequences of user actions (Hammond, 1987).

Control Compatibility

Some control movements are easier to learn than others, and this depends heavily on prior experience. If you move a computer mouse to the right, causing an object on the screen to move to

the right, this positioning method will be easy to learn. A skill is being applied that was gained very early in life and has been refined ever since. But if the system interface has been created such that a mouse movement to the right causes a graphical object to move to the left, this will be incompatible with everyday experience and positioning the object will be difficult. In the behaviorist tradition of psychology, this factor is generally called *stimulus–response (S–R) compatibility*. In modern cognitive psychology, the effects of S–R compatibility are readily understood in terms of skill learning and skill transfer.

In general, it will be easier to execute tasks in computer interfaces if the interfaces are designed in such a way that they take advantage of previously learned ways of doing things. Nevertheless, some inconsistencies are easily tolerated, whereas others are not. For example, many user interfaces amplify the effect of a mouse movement so that a small hand movement results in a large cursor movement. Psychologists have conducted extensive experiments that involve changing the relationship between eye and hand. If a prism is used to laterally displace what is seen relative to what is felt, people can adapt in minutes or even seconds (Welch and Cohen, 1991). This is like using a mouse that is laterally displaced from the screen cursor being controlled.

On the other hand, if people are asked to view the world inverted with a mirror, it can take weeks of adaptation for them to learn to operate in an upside-down world (Harris, 1965). Snyder and Pronko (1952) had subjects wear inverting prisms continuously for a month. At the end of this period, reaching behaviors seemed error-free, but the world still seemed upside-down. This suggests that if we want to achieve good eye–hand coordination in an interface, we do not need to worry too much about matching hand translation with virtual object translation, but we should worry about matching the axis or rotation.

Some imaginative interfaces designed for virtual reality involve extreme mismatches between the position of the virtual hand and the proprioceptive feedback from the user's body. In the Go-Go Gadget technique (named after the cartoon character, Inspector Gadget), the user's virtual hand is stretched out far beyond his or her actual hand position to allow for manipulation of objects at a distance (Poupyrev et al., 1996).

Studies by Ramachandran (1999) provide interesting evidence that even under extreme distortions people may come to act as if a virtual hand is their own, particularly if touch is stimulated. In one of Ramachandran's experiments, he hid a subject's hand behind a barrier and showed the subject a grotesque rubber Halloween hand. Next, he stroked and patted the subject's actual hand and the Halloween hand in exact synchrony. Remarkably, in a very short time, the subject came to perceive that the Halloween hand was his or her own. The strength of this identification was demonstrated when the researcher hit the Halloween hand with a hammer. The subjects showed a strong spike in galvanic skin response (GSR), indicating a physical sense of shock. No shock was registered without the stroking. The important point from the perspective of virtual reality interfaces is that even though the fake hand and the subjects' real hand were in quite different places, a strong sense of identification occurred.

Consistency with real-world actions is only one factor in skill learning. There are also the simple physical affordances of the task itself. It is easier for us to make certain body movements

than others. Very often we can make computer-mediated tasks easier to perform than their real-world counterparts. When designing a house, we do not need to construct it virtually with bricks and concrete. The magic of computers is that a single button click can often accomplish as much as a prolonged series of actions in the real world. For this reason, it would be naïve to conclude that computer interfaces should evolve toward VR simulations of real-world tasks or even enhanced Go-Go Gadget–style interactions.

Vigilance

A basic element of many interaction cycles is the detection of a target. Although several aspects of this have already been discussed in Chapter 5, a common and important problem remains to be covered—the detection of infrequently appearing targets.

The invention of radar during World War II created a need for radar operators to monitor radar screens for long hours, searching for visual signals representing incoming enemy aircraft. Out of this came a need to understand how people can maintain *vigilance* while performing monotonous tasks. This kind of task is common to airport baggage X-ray operators, industrial quality-control inspectors, and the operators of large power grids. Vigilance tasks commonly involve visual targets, although they can be auditory. There is extensive literature concerning vigilance (see Davies and Parasuraman, 1980, for a detailed review). Here is an overview of some of the more general findings, adapted from Wickens (1992):

1. Vigilance performance falls substantially over the first hour.
2. Fatigue has a large negative influence on vigilance.
3. To perform a difficult vigilance task effectively requires a high level of sustained attention, using significant cognitive resources. This means that dual tasking is not an option during an important vigilance task. It is not possible for operators to perform some useful task in their "spare time" while simultaneously monitoring for some difficult-to-perceive signal.
4. Irrelevant signals reduce performance. The more irrelevant visual information is presented to a person performing a vigilance task, the harder it becomes.

Overall, people perform poorly on vigilance tasks, but there are a number of techniques that can improve performance. One method is to provide reminders at frequent intervals about what the targets will look like. This is especially important if there are many different kinds of targets. Another is to take advantage of the visual system's sensitivity to motion. A difficult target for a radar operator might be a slowly moving ship embedded in a great many irrelevant noise signals. Scanlan (1975) showed that if a number of radar images are stored up and rapidly replayed, the image of the moving ship can easily be differentiated from the visual noise. Generally, anything that can transfer the visual signal into the optimal spatial or temporal range of the visual system should help detection. If the signal can be made perceptually different or distinct from irrelevant information, this will also help. The various factors that make color, motion, and texture distinct can all be applied. These are discussed in Chapters 4 and 5.