Chapter 1

Graphics Input, Storage, and Communications

1.1 GRAPHICS INPUT DEVICES

Any device that allows information from outside the computer to be communicated to the computer is considered an *input device*. Since the central processing unit (CPU) of a digital computer can understand only discrete binary information, all computer input devices and circuitry must eventually communicate with the computer in this form. Many devices are capable of performing this task. Some common computer graphics input devices are:

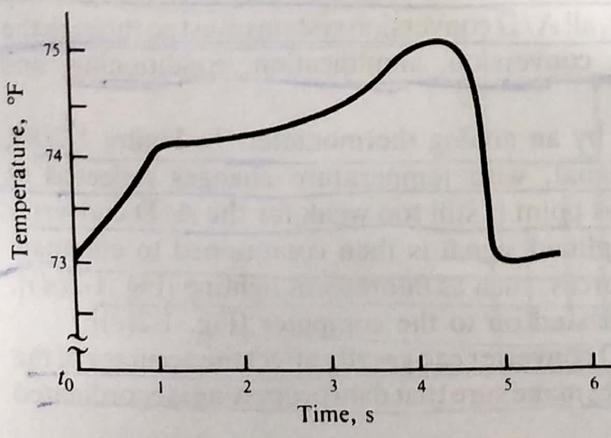
- Keyboard
- Trackball
- Joystick
- Mouse
- Paddle controls
- Light pen
- Magnetic tablet
- Digitizing camera

Analog Devices versus Digital Devices

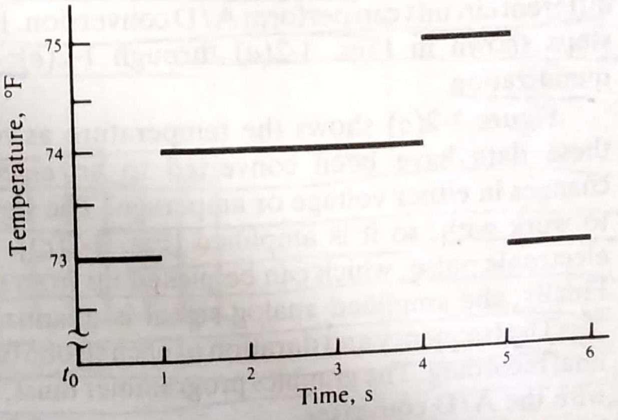
Input devices are of two basic types, *analog* and *digital*. This section provides a brief overview and explanation of how some of the more popular analog and digital devices work.

Humans perceive a continuous universe; digital devices sense a discrete universe consisting of discrete segments. For example, the temperature indicated by a digital thermometer may be observed to change from 73 to 74 degrees instantaneously. We know that the actual temperature did not suddenly jump from 73 to 74 degrees. Rather, there was a continuous warming of the air, and at some moment the temperature became closer to 74 than 73 degrees. If a more accurate digital thermometer were used such as one that could measure tenths of a degree, the exact temperature would still not be sensed. Instead, closer approximations would be given.

Figures 1-1(a) and 1-1(b) illustrate the differences in the way temperatures would be recorded by a



(a)



(b)

Fig. 1-1

continuous [Fig. 1-1(*a*)] and a discrete [Fig. 1-1(*b*)] device. Notice that while the temperature is actually changing continuously, the discrete device only records incremental changes of one or more degrees. The device capable of recording continuous changes is an analog device; the device which can sense only discrete changes is a digital device.

Time is also perceived as continuous. (However, recent physics experiments have begun to challenge this concept.) Therefore, a digital device must take "snapshots" of the environment with sufficient rapidity if a high degree of accuracy is to be attained. When a computer takes a snapshot of the environment by obtaining a value from an input device, it is said to *strobe* the device. For example, Table 1-1 demonstrates the temperature readings that would be recorded 3.5 seconds (s) from time t_0 if a digital device accurate to one degree were used to record the temperatures shown in Fig. 1-1(*a*); the frequency of strobing ranges from once every 0.5 s to once every 5 s starting at t_0.

Table 1-1 Effects of Strobe Frequency on Accuracy

Frequency of Strobe	Observed Temperature at 3.5 s from t_0
0.5	75
1.0	75
2.0	74
3.0	74
4.0	73
5.0	73

electret

Analog-to-Digital Conversion

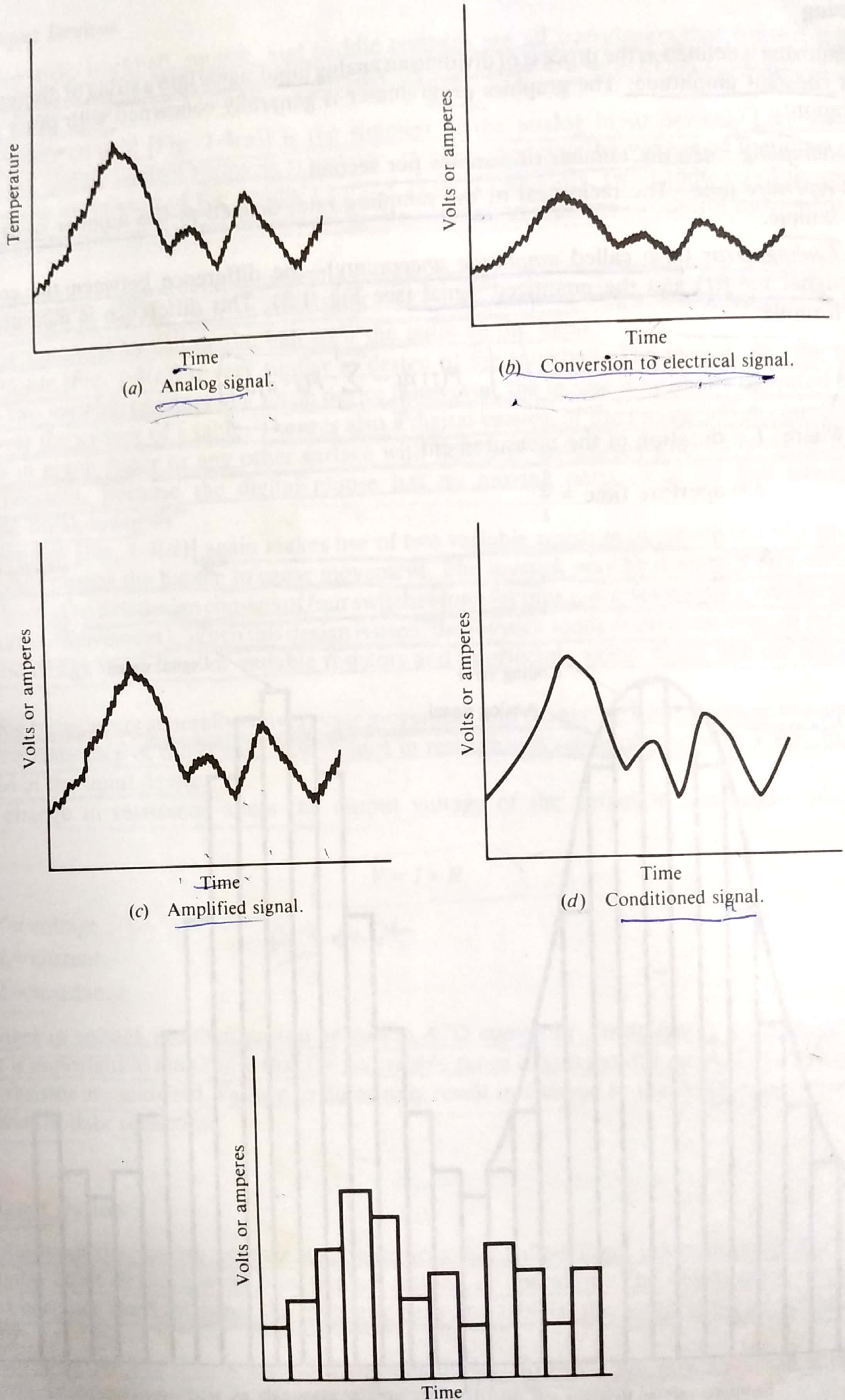
Since the computer is a discrete digital device, while most data in the world outside the computer is continuous, the data must be converted to enable the computer to use it. The conversion of continuous analog data to discrete digital data is called *analog-to-digital conversion*, or simply *A/D conversion*.

The computer graphics programmer must have some knowledge of A/D conversion because many graphics input devices are analog devices and A/D conversion can affect the accuracy of data collected even while exhibiting a high degree of precision.

Typically, A/D conversion is performed either by hardware devices incorporated in the computer's input circuitry or by means of firmware that directly interfaces with the computer's input circuitry. Many different circuits can perform A/D conversion. However, all A/D conversion systems must go through the steps shown in Figs. 1-2(*a*) through 1-2(*e*): sensing, conversion, amplification, conditioning, and quantization.

Figure 1-2(*a*) shows the temperature as recorded by an analog thermometer. In Figure 1-2(*b*), these data have been converted to an electronic signal, with temperature changes reflected as changes in either voltage or amperage. The signal at this point is still too weak for the A/D converter to work with, so it is amplified [Fig. 1-2(*c*)]. The amplified signal is then conditioned to eliminate electronic noise, which can be picked up from many sources, such as fluorescent lighting [Fig. 1-2(*d*)]. Finally, the amplified analog signal is quantized and passed on to the computer [Fig. 1-2(*e*)].

The frequency and duration of each strobe of the A/D converter can greatly affect the accuracy of the final recording. The graphics programmer must, therefore, make sure that data processing is coordinated with the A/D converter.



(*a*) Analog signal.

(*b*) Conversion to electrical signal.

(*c*) Amplified signal.

(*d*) Conditioned signal.

(*e*) Quantized signal.

Fig. 1-2

Quantizing

Quantizing is defined as the process of dividing an analog input signal into a string of discrete outputs, each of constant amplitude. The graphics programmer is generally concerned with three aspects of quantization:

1. *Sampling rate*—the number of samples per second.
2. *Aperture time*—The reciprocal of the sampling rate, defined as the number of seconds per sample.
3. *Coding error* (also called *amplitude uncertainty*)—the difference between the conditioned signal $y = f(t)$ and the quantized signal (see Fig. 1-3). This difference is measured by the formula

$$\int_0^T F(t)\, dt - \sum_{i=0}^{N-1} F(i \cdot a)a \tag{1.1}$$

where T = duration of the measurement

a = aperture time $= \dfrac{1}{s}$

$N = \dfrac{T}{a}$

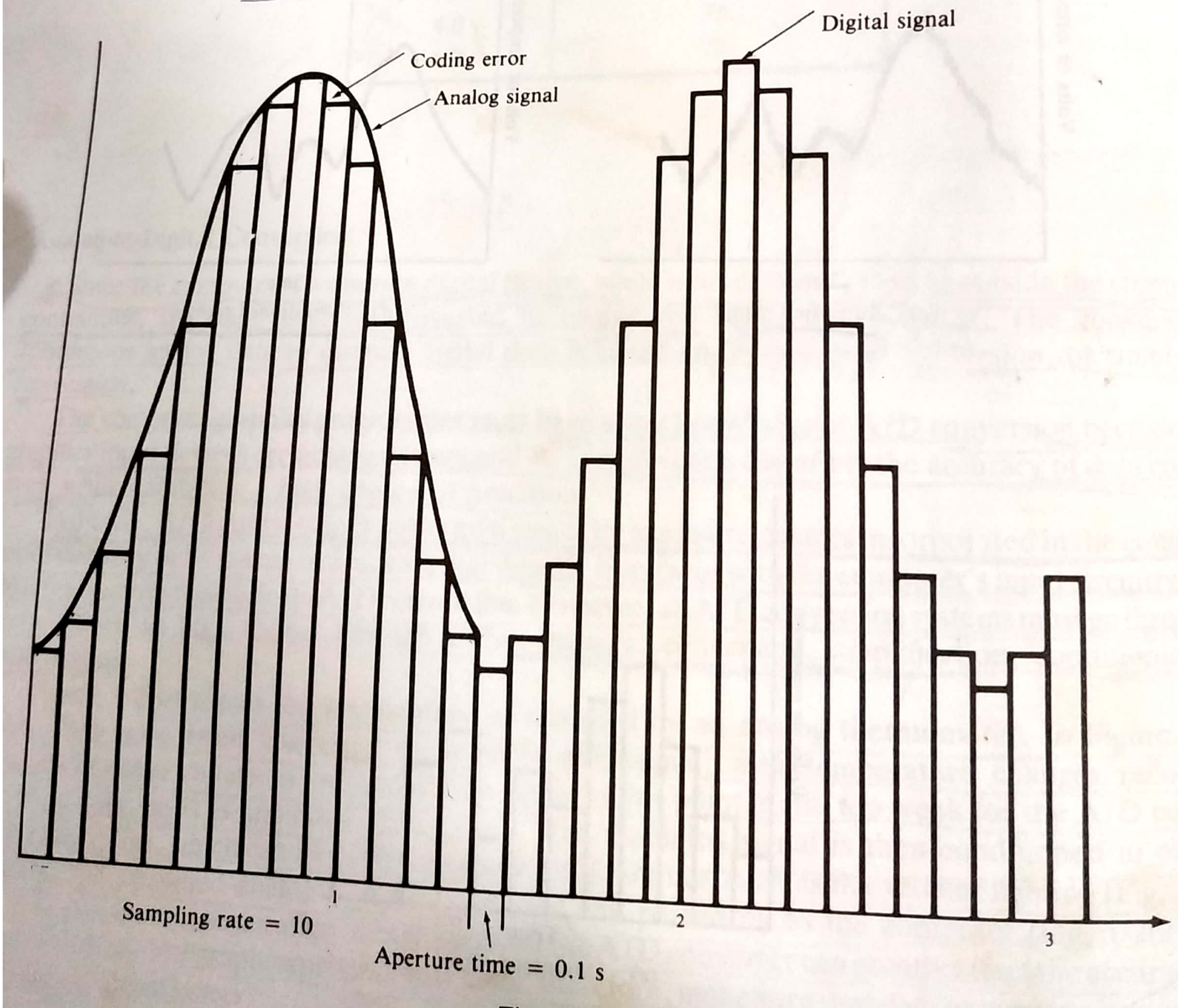


Fig. 1-3

Transducer: That converts energy from one form to another

Analog Input Devices

The joystick, trackball, mouse, and paddle controls are all transducers that convert a graphics system user's movements into changes in voltage. A *transducer* is a device that converts energy from one form to another.

The *paddle control* [Fig. 1-4(*a*)] is the simplest of the analog input devices. Like all variable resistors, the paddle control varies its resistance, thereby changing the voltage of the input circuit in relation to the movement of the paddle's control knob. Commonly, two paddle controls are used in graphics systems, one to control movement in the x direction and one to control movement in the y direction.

The *trackball* [Fig. 1-4(*b*)] mechanically combines two variable resistors in a single device, thus allowing the user to use one hand to enter both x and y information with a single device. The trackball is normally operated by rolling the ball with the palm of the hand.

The *mouse* [Fig. 1-4(*c*)] is very similar in design to the trackball. The mouse, like the trackball, combines two variable resistors in a single device. However, the mouse is normally operated by rolling the ball over the surface of a table. There is also a digital version of the mouse which counts light and dark lines in graph paper or any other surface with light and dark lines, thus eliminating the use of variable resistors. Because the digital mouse has no moving parts, it is far less susceptible to mechanical breakdown.

The *joystick* [Fig. 1-4(*d*)] again makes use of two variable resistors to specify x and y movement. Here the user pushes the handle to cause movement. The joystick may be designed with either of two basic methods. The first design consists of four switches (one for plus x, one for minus x, one for plus y, and one for minus y movement). When this design is used, the joystick sends relative movement information. The second design makes use of variable resistors and in principle works much like the trackball and mouse.

Analog transducers generally convert user movement into changes in either voltage or amperage by varying the resistance of the circuit. The change in resistance is accomplished with a variable resistor embedded in the input device.

The change in resistance alters the output voltage of the circuit in accordance with Ohm's law:

$$V = I * R \tag{1.2}$$

A] impt

where V = voltage
I = current
R = resistance

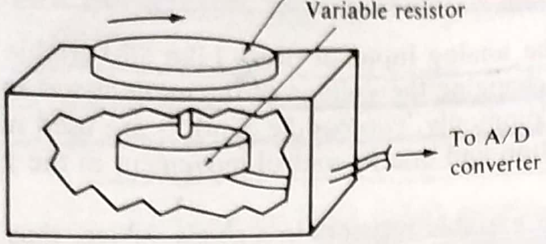
The changes in voltage are then passed on to the A/D converter. In designing a computer graphics system it is important to make sure that the acceptable range of voltages for the A/D converter and the variable resistor are matched. Failure to do so may result in damage to the A/D converter or precise but inaccurate data collection.

Digital Input Devices

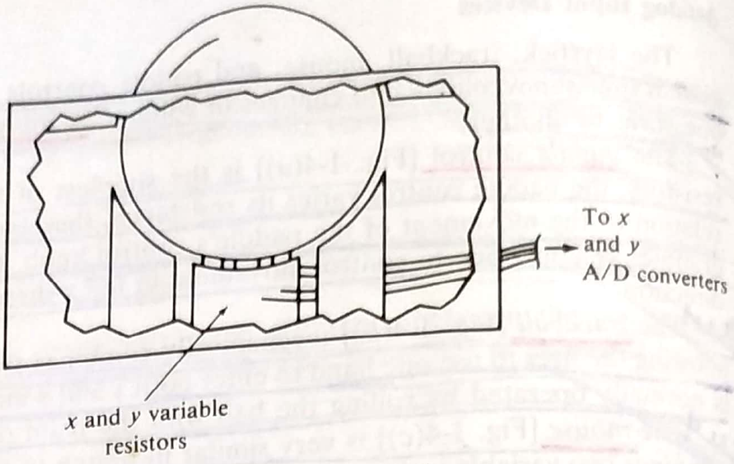
Digital input devices are actually analog devices that collect input information in discrete form. Each digital input device has its own distinct method of operation. The digital input devices to be analyzed here are the light pens, the magnetic pens and tablets, the touch-sensitive screen, and the keyboard.

A *light pen* (see Fig. 1-5) is used as a pointing device. Typically the user will point with the pen to perform an operation such as drawing a line or rotating an object on a CRT.

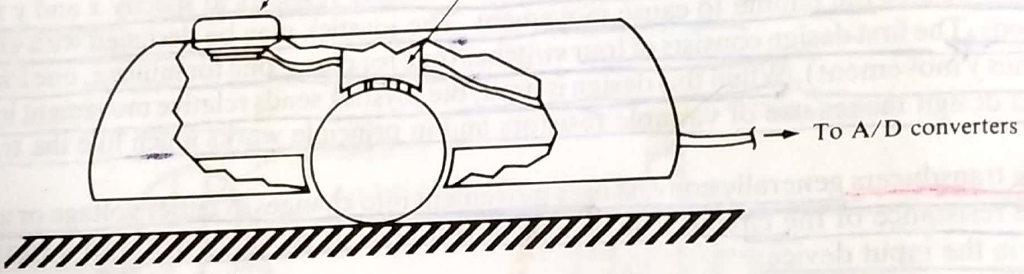
Although many light pen systems emit light from the tip of the pen, the location to which the user is pointing is not found by light sensors in the CRT. Instead, the light pen senses the light emitted by the CRT.



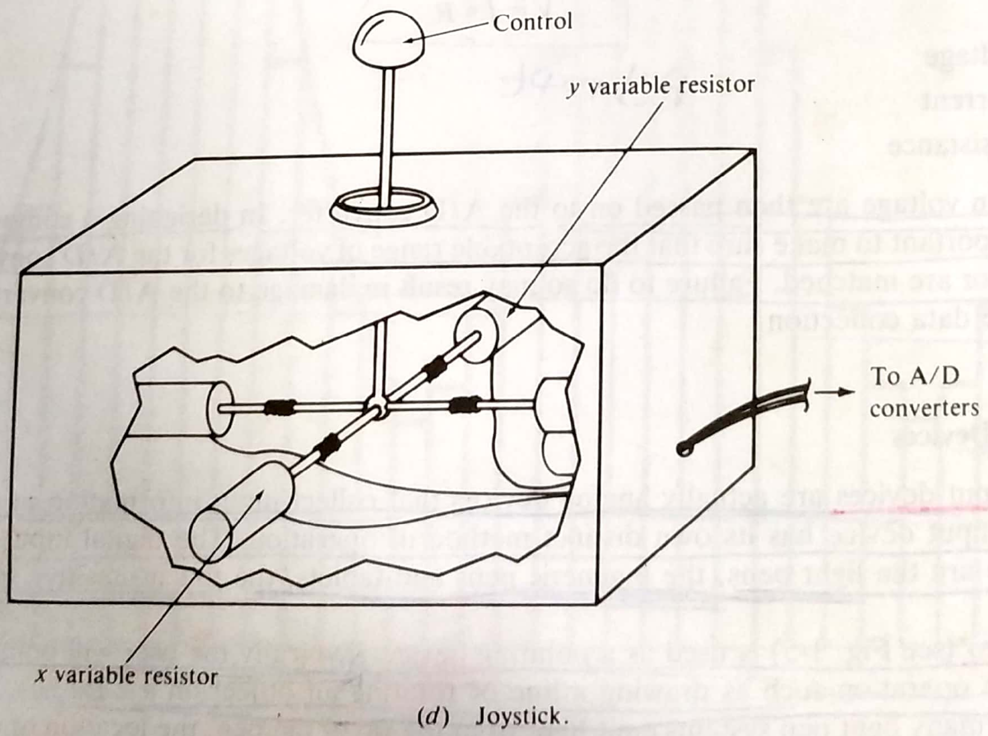
(*a*) Paddle control.



(*b*) Trackball.



(*c*) Mouse.



(*d*) Joystick.

Fig. 1-4

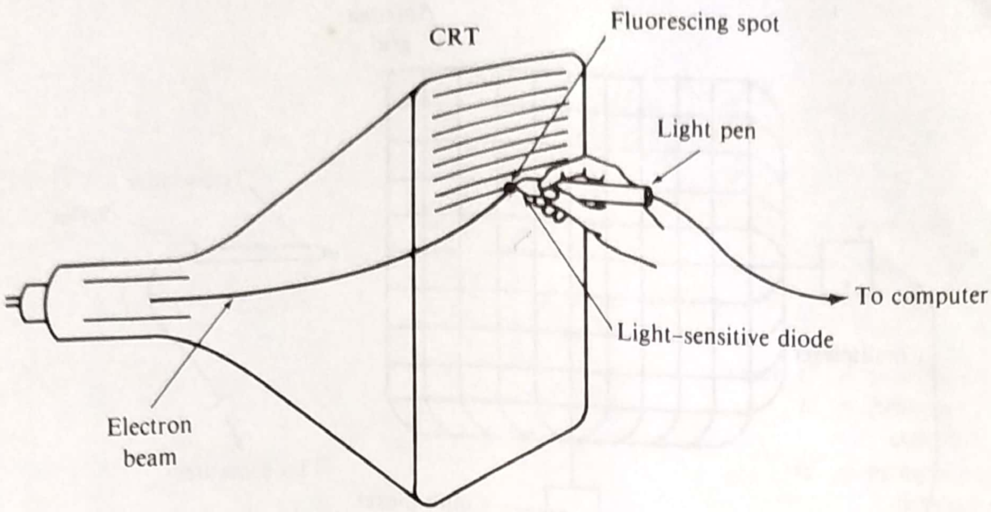


Fig. 1-5 Light pen system.

The location to which the user is pointing can be found by first keeping track of when the CRT's electron beam starts each refresh cycle. Then, the amount of time that passes before the light pen receives a pulse of light (indicating that the beam has just passed) is measured. When a noninterlaced CRT is used, the duration of each refresh cycle will normally be equal to the line frequency [60 hertz (Hz) in the United States and 50 Hz elsewhere].

As discussed in Chap. 2, the computer "perceives" the pixels' ouput to the CRT as a continuous line. Therefore, the percentage of the refresh cycle completed before the light pen receives a pulse should be equal to the percentage of the screen that has been written to; this, in turn, should be equal to the percentage of display memory that has been written. Because of overscan, however, the task of locating where the user is pointing becomes far more complex. The problem is further complicated when an interlaced CRT is used.

The computer does not have to actually time the period between the start of each refresh cycle and the time when a pulse is recorded by the pen because the computer is already keeping track of which portion of display memory is being written at any given moment. Therefore, the computer can easily locate where the user is pointing by storing the address of the byte which is being displayed at the time the light pen records a pulse (see Prob. 1.12).

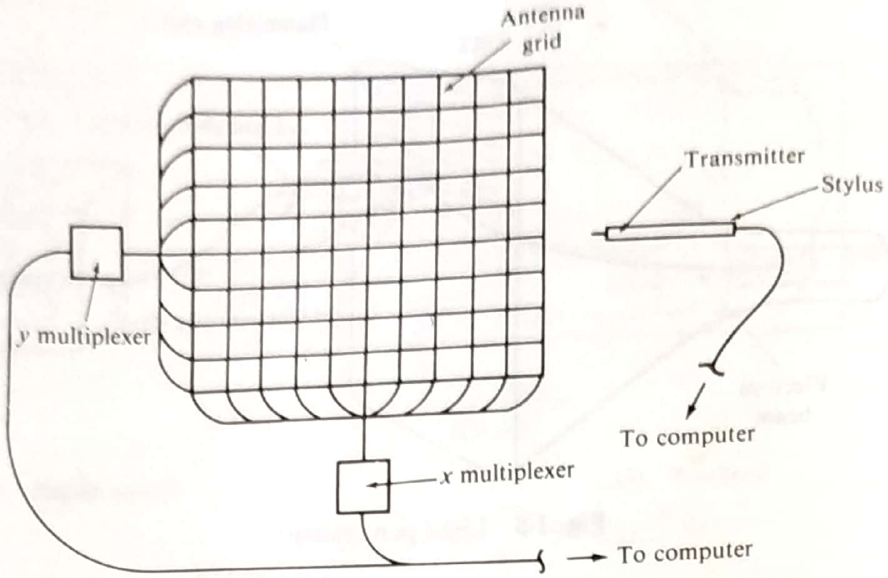
A *magnetic pen* and *tablet* [see Fig. 1-6(*a*)] are composed of a two-dimensional wire grid and a radiowave-emitting stylus. The wire grid is a matrix antenna which locates the position of the stylus measuring the intensity of the radio signal received by each wire in the grid. By comparing the intensity of signal received by each wire in the grid the demultiplexing circuitry, you will be able to calculate the position of the stylus even when it lies between wires.

An *acoustic tablet* [see Fig. 1-6(*b*)] makes use of strip microphones located around the perimeter of the tablet. The stylus generates a series of small sparks when placed near the tablet. The acoustic tablet has been losing popularity, however, because it is too noisy (it generates a buzzing sound) and too susceptible to ambient noise. Capacitance tablets are also quite popular. The capacitance tablet operates the same way as the touch-sensitive screen described next.

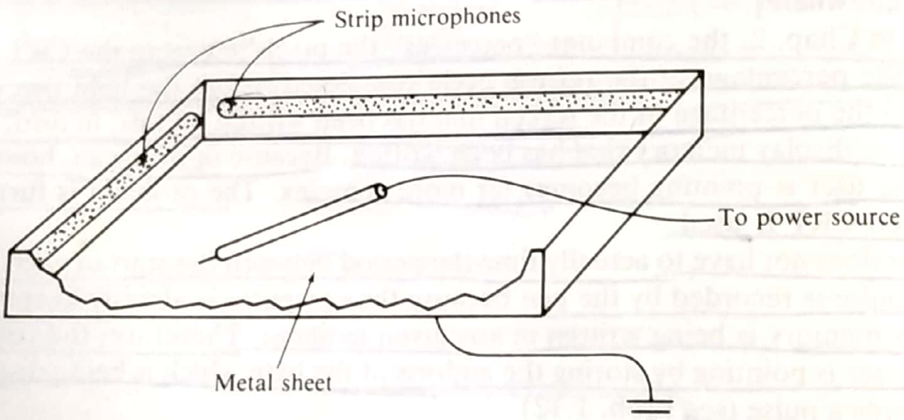
There are three basic types of *touch-sensitive screens*. The first design incorporates a fine grid of wire pairs placed over the face of the display. When the user presses the screen, the capacitance of the wire pairs is altered, thus indicting the position of the user's finger [see Fig. 1-7(*a*)].

The second design utilizes a series of strain gauges located around the perimeter of a plate of glass or plastic placed over the display. When the user presses the plate, the material is deformed, thereby allowing the location of the user's finger to be calculated [see Fig. 1-7(*b*)].

strain gauges



(a) Magnetic tablet.



(b) Acoustic tablet.

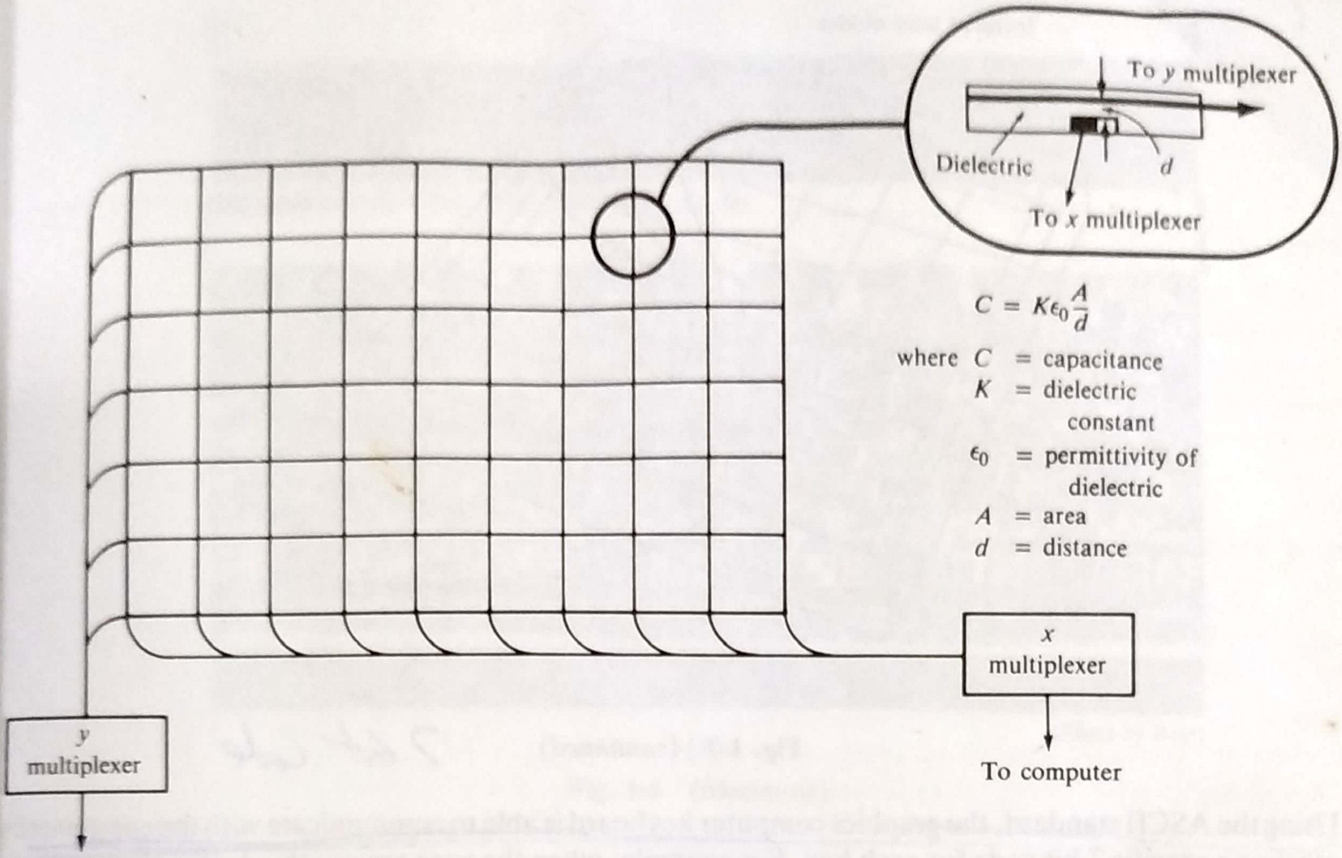
Fig. 1-6

The most popular method of creating a touch-sensitive screen makes use of a series of infrared light-emitting diodes (LEDs) and sensors located around the perimeter of the display [see Fig. 1-7(c)]. When the user touches the screen, light beams are broken, indicating the location of the user's finger.

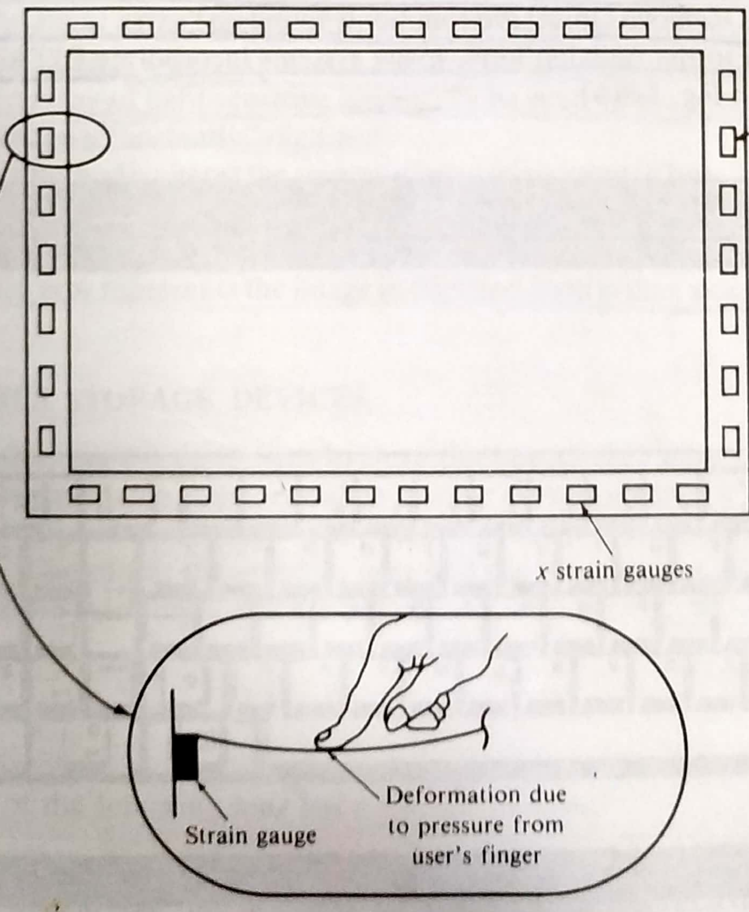
Touch-sensitive screens, although popular in concept, have been slow to be implemented because of technical problems. For example, in the first method, the wire grid covering the screen has been found to be too susceptible to wear. In the second method, the plate is too easily affected by changes in temperature (heat causes the plate to expand, thus causing strain) and mechanical fatigue. In the third method, designers have had trouble creating a sufficiently dense matrix of light beams. As a result, the user's finger is often placed either between the light beams or in a location in which only one beam is broken.

The *keyboard* is a common input device. A standard known as the American Standard Code for Information Interchange (ASCII) has been developed to allow computers to encode keyboard characters.

ASCII

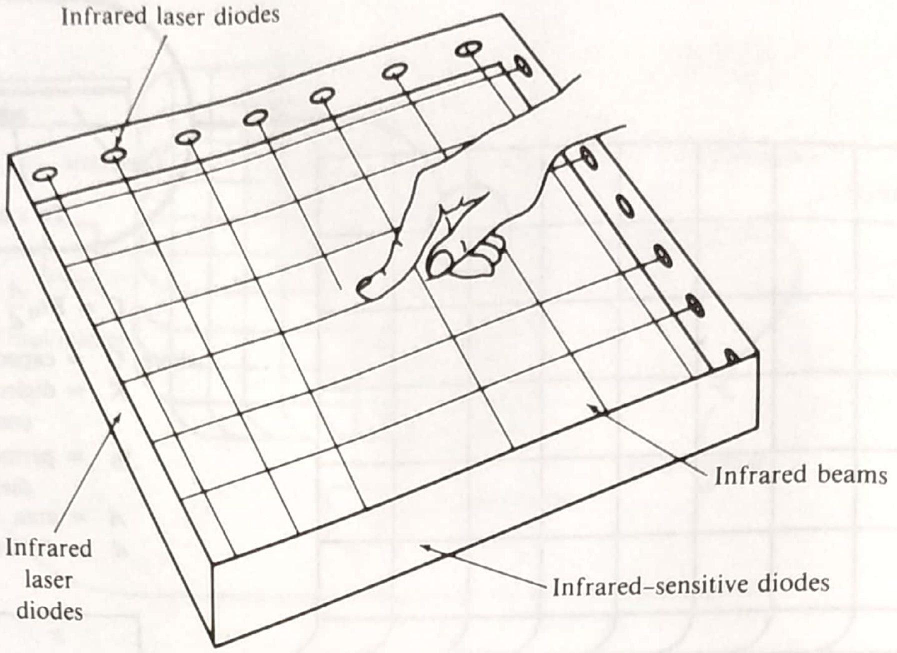

To y multiplexer
Dielectric
d
To x multiplexer
$C = K\epsilon_0 \frac{A}{d}$
where C = capacitance
K = dielectric constant
ϵ_0 = permittivity of dielectric
A = area
d = distance
x multiplexer
To computer
y multiplexer
To computer

(a)


y strain gauges
x strain gauges
Strain gauge
Deformation due to pressure from user's finger

(b)

Fig. 1-7

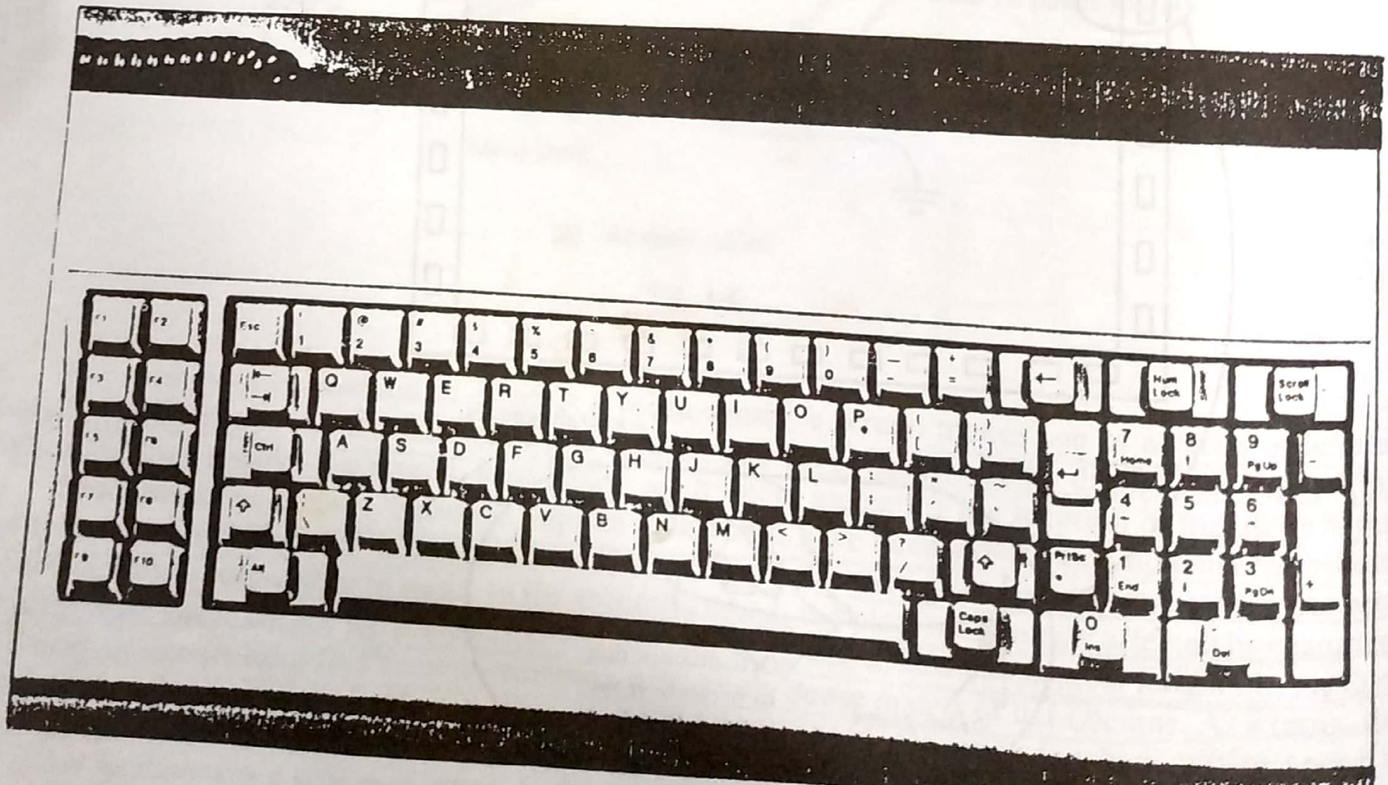


(*c*)

Fig. 1-7 (*continued*)

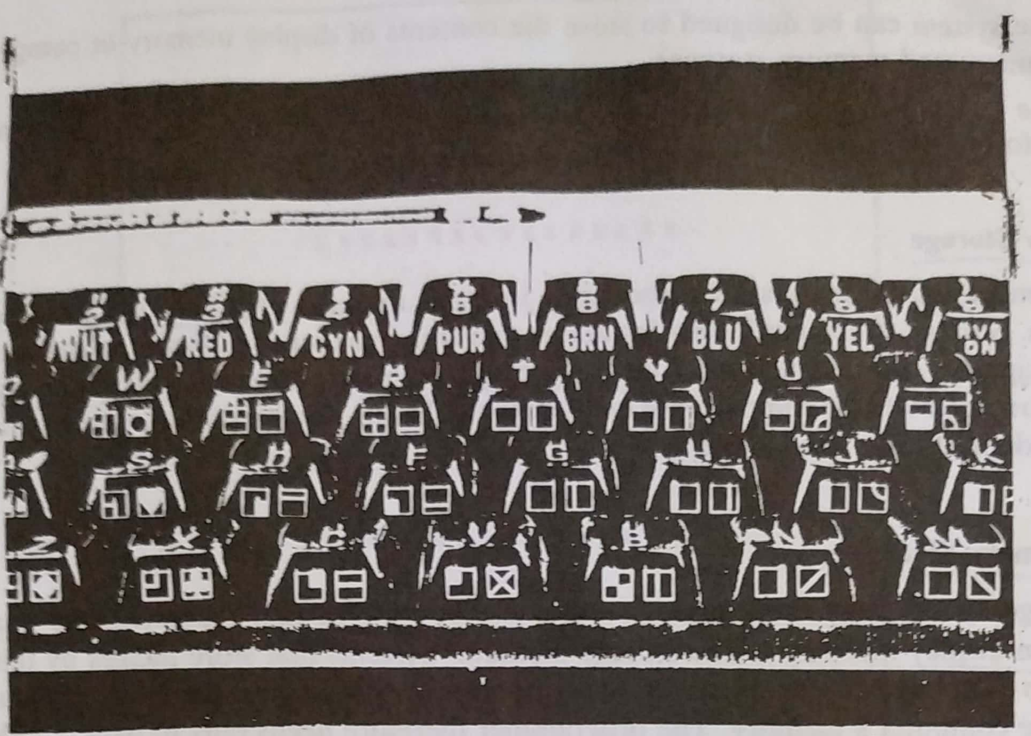
7 bit Code

Using the ASCII standard, the graphics computer keyboard is able to communicate with the computer by sending a specific 7-bit code for each key. For example, when the user presses the A key, the computer generally will receive the ASCII for the letter "A" (the entire ASCII character set is shown in App. 3). Many graphics system keyboards produce more than the standard 128 ASCII character set. Functions such as Save Drawing, Rotate, and Remove Object are commonly found on "extra" keys called *function keys* [see Fig. 1-8(*a*)]. In addition to the function keys, many systems incorporate BITBLT (see Chap. 3) graphics symbols, as shown in Fig. 1-8(*b*).



(*a*)

Fig. 1-8



(Photo by Boto)

(b)

Fig. 1-8 *(continued)*

Digital Cameras and Digitizing Video Images

Images can be converted into digital form. The video image to computer transformation is usually accomplished in either of the following two ways. In the first method, a lens is used to focus light on a bank of light-sensitive diodes. When exposed to sufficiently high light intensity, a diode will jump into the "on" state. Since the array of light-sensitive diodes can be read by the computer as random-access memory (RAM), the image is "instantly" digitized.

The second method is almost the reverse of scan conversion (Chap. 3). The video signal generated by a video camera represents a series of scan lines. With the use of an A/D converter the signal is "chopped" into pixel-sized segments. (The level of each segment determines whether a pixel is on or off.) The resulting bit stream which now represents the image in digitized form is then stored in a buffer for later processing.

1.2 GRAPHICS STORAGE DEVICES

Most graphics system users want to store the images they have created. There are many storage devices currently available. Some common storage devices are floppy disks, hard disks, magnetic tape, paper tape, punched cards, video tape, and film. Each storage device has unique advantages and disadvantages. Therefore, the graphics user and the programmer should carefully examine the final application before selecting a storage device.

Graphics Storage Formats

Regardless of the storage medium selected, the graphics system designer will always use some combination of the following four basic storage formats:

1. The system can be designed to store only the image(s) created (image-only storage).
2. The system can be designed to store bit by bit exactly what is in display memory (display-memory storage).