

In providing background information on display technology, this paper discusses cathode-ray tube characteristics and interactive devices as they affect the user.

Described are the display functions of the IBM 2250 display console, which is used in many applications.

Some elementary aspects of image generation are presented, and the current and potential capability of displays is discussed.

INTERACTIVE GRAPHICS IN DATA PROCESSING

Aspects of display technology

by **A. Appel, T. P. Dankowski, and R. L. Dougherty**

Immediate computer reaction to an operator's actions at a cathode-ray tube display console was a long sought-after goal that was successfully achieved during the nineteen fifties in such military applications as the SAGE air defense system.¹ Of course, SAGE also required programming for interactive graphic operations. Developments such as this struck several investigators at about the same time with the potential for broader uses of intelligent displays. By the early nineteen sixties, academic and industrial investigators had developed both the equipment and programming^{2,3,4} that inspired predictions of a new millenium in computer usage.

During the early developments, it was assumed that all problems in computer graphics could eventually be solved by human intelligence through programming and that available digital and analog equipment would meet programming demands. However, solutions of programming problems in interactive languages,⁵ systems modeling,⁶ information retrieval,⁷ and machine drafting^{8,9} now indicate that progress may continue to be determined by improvements in computer graphics equipment, in both quality and variety.

For a better understanding of current developments in this area and as background information for the programming-oriented papers of this graphics issue pertinent aspects of display technology are discussed in this paper—not so much from the designer's point of view, but rather as they affect the console user. An elaboration on IBM 2250 display console considerations is preceded

by more general aspects of cathode-ray tube (CRT) display characteristics and interactive devices. Some of the elementary aspects of image generation are discussed, and finally some comments are made on the current and potential capabilities of displays.

Device concepts

Display devices for computer graphics comprise mechanical displays (e.g., a computer-driven pen) and CRT displays. This paper is restricted to a discussion of displays of the CRT type, which may be further classified as video and digital. Video displays use television techniques and are generally restricted to performing output functions. On digital displays, the electron beam is positioned and moved sequentially from point to point. Here we are primarily concerned with digital display-generating devices, which can perform input as well as output functions.

Digital display consoles use computer-controlled x - y positioning of the CRT electron beam to draw images on the screen. This image-generating technique differs from the raster scan technique used for television in which the video scan simulates a picture by a series of unblanked (luminous) points on the raster. For example, many alphanumeric displays are simply television outputs wherein the electron beam is directed through a mask into which the alphanumeric characters have been pierced. In this mode, the information required is the character name (e.g., "1," "2," "A," "B," etc.), line number, and character position in the line. Of course, typical displays of this type contain many lines of many characters each. For example, depending upon the controller, the IBM 2260 display console is capable of producing 240, 480, or 960 characters per display.

Advantages of video display are: greater range of tone, simpler storage (e.g., analog tapes), low cost, ease of manufacturing, and the ability to use existing video transmission facilities. The disadvantage of video display is that it is difficult to interact with graphic elements larger than a single point.

In this paper, we are particularly interested in discussing display devices that generate continuous curves by a series of closely spaced points or connected line segments. Advantages of a line-generating CRT derive from the fact that the lines used to generate the picture are in a rationally organized display list, and this list can be easily processed by a computer. For example, we can detect specific lines or line groups (pictures) to which they belong. Thus, we can choose specific graphic elements on the screen. A disadvantage of the line-generating CRT is the manufacturing precision required for accuracy and stability of the graphic input and output.

Line-generating graphics devices require continuous regeneration of the display data at rates from 15 to 60 times per second (depending on the persistence of the CRT phosphor) to avoid flickering of the picture on the screen. The "flickering problem" in-

CRT
characteristics

herently limits the amount of information displayed regardless of size of storage available. As the amount of information to be displayed increases, the total movement of the electron beam increases until the time required to make one generating sweep of the display list exceeds the persistence of the phosphor. At this point, the display starts to flicker and becomes uncomfortable to view. Regeneration is normally performed with a separate storage buffer (such as core or drum) so that the central processing unit (CPU) is free to perform arithmetic computations. This buffer can be segmented and shared by multiple displays. For a small computer, however, it can be more economical to allow the CPU main storage to perform the dual role of storing the display data for regeneration and the application program when only one display console is attached.

Storage tubes are an economical solution to the problem of flicker-free line display without a large buffer. The surface of the CRT "remembers" what was displayed, and does not have to be continuously regenerated. The disadvantage of present storage CRT devices is the long write time and the inability to selectively erase. The time required to erase and to regenerate a picture makes the storage tube impracticable for most dynamic display applications.

Analog input devices (such as dials and control sticks) can be directly manipulated by the user to change stored values in the buffer. Analog devices can be added to the basic display to directly produce linear transformations such as three-dimensional to two-dimensional projections, and rotation and translation of an object in space. The analog perspective-generating devices¹⁰ can be cited as an example.

interactive
devices

Use of the CRT as the output medium of a graphic display console for interactive systems has become almost standard; however, no device is considered as being in a similar position in providing the interactive input functions from these consoles. The variety of input devices range from simple function keys, which merely interrupt the computer and present the key number, to complex position-detecting mechanisms such as the RAND tablet.^{11,12} For direct interaction on a line-generating CRT screen, a light pen can be used to detect displayed elements. Another class of input devices provides continuous (analog) information to the program. Examples include potentiometers, "joy sticks," "tracking balls," and "force sticks," all of which are hand-manipulated to provide positional, angular, or other data that the program can use, for example, to position a cursor (small symbol used for pointing on display) or to rotate or translate an image. These devices give the user a strong physical association with the displayed image by allowing cursor movement, figure rotation, etc., to "follow" the motion of his hand. More random movement is permitted, for example, in positioning of the cursor, whereas a cursor from an alphanumeric keyboard is restricted to rigid vertical and horizontal motions and is usually locked into the alphanumeric mode of display regeneration.

The alphanumeric keyboard, an input device available on many displays, has a wide variety of computer interrupt and text-editing features in addition to the standard alphanumeric keys. Some displays that have programmable character sets can offer keys and type fonts to suit the user. For instance, the IBM 2250 Model 4 display console offers both upper- and lower-case character capability.

In spite of the wide variety of input devices, it frequently requires much effort to describe the geometry of an object. For example, if we want to encode a very detailed drawing, we have to do almost as much work as was involved in making the drawing initially. Another problem is the dimensionality of input, i.e., describing an object or shape that changes in a complex way with time (such as animation).

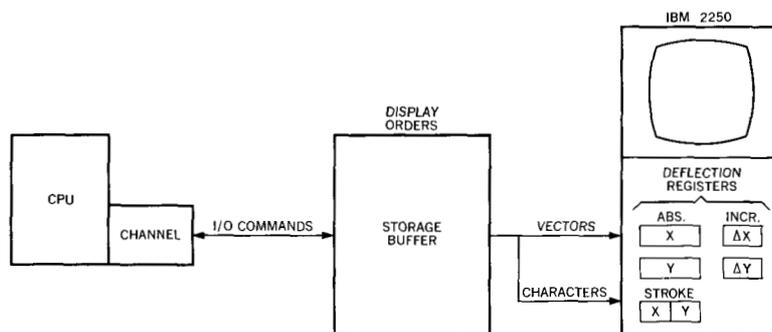
The 2250 display console provides the features required to satisfy most interactive applications, and is easily handled within SYSTEM/360 architecture. However, there is always a need for display devices with specialized graphics features to meet the unique requirements imposed by some users. In these cases, application-oriented programming support can greatly increase the utility of any display console and possibly satisfy the functions of special features.

Display technology of the IBM 2250

In discussing detailed display functions, the IBM 2250 display console—used in several applications described in this issue—serves as an example of commercially available display devices.

An image on a 2250 is initiated by an input/output command from the CPU, and regeneration continues under control of orders accessed from buffer storage, as shown in Figure 1. The display orders form a stored program which is executed in a manner similar to a CPU-stored program. Functions performed by this order program include vector generation, character generation, light-pen control, branching, subroutines, logical decisions, timing, and interrupt generation.

Figure 1 Generation of an image on the IBM 2250



Vector generation is accomplished by providing, through orders, the end points of each vector either as absolute coordinates or as relative increments which are added to the coordinates of the preceding end point. Points may also be plotted in the same manner. Characters are generated by using a series of short vectors or strokes. These strokes may be machine-generated from the normal eight-bit CPU representation of a character, or specified by stroke tables for each character in buffer storage; this is a 2250 model-dependent function.

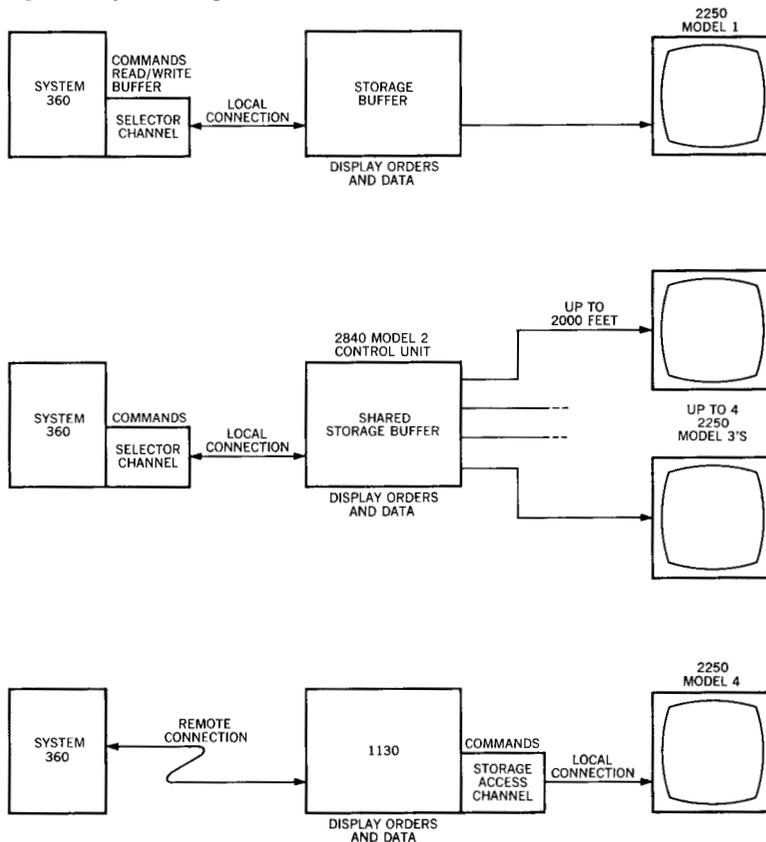
The 2250 Model 3 and Model 4 allow for linkages to graphics order subroutines. This function, in combination with the relative vector orders, permits an object appearing in multiple locations on the screen to be stored only once in buffer storage. Each time the subroutine is called, it draws the object in a relocated position relative to the current absolute contents of the x - y deflection registers. Logical decisions based on pen switch mode may also be performed in the 2250 without intervention by the processor. Also provided are conditional branch and conditional interrupt orders to test for deferred light-pen actions.

Figure 2 indicates the variations available in system configurations with the 2250. The 2250 Model 1 has a self-contained storage buffer of up to 8,192 bytes with a cycle time of four microseconds per byte. The display orders, such as ENTER VECTOR MODE, ENTER POINT MODE, ENTER CHARACTER MODE, and the data indicating x - y positions of lines and characters are stored in digital form in this buffer for display regeneration. The digital position data is converted to analog form before being applied to the deflection coils of the CRT for beam positioning. The beam can be turned off or on for each movement to create any desired image on the screen. The Model 1 connects to a selector or multiplexor channel on an IBM SYSTEM/360, and the storage buffer contents can be written out of main storage and read into main storage by such commands as READ/WRITE buffer.

The 2250 Model 3 shares a large storage buffer (32,768 bytes at two microseconds for two bytes), which is in a separate display control unit, with up to three additional Model 3 display consoles. The allocation of space in this buffer is completely flexible under program control. The organization of commands, display orders, and data is the same as for the Model 1, except the Model 3 has an extended set of orders. The additional orders can be used to perform light-pen tracking in the buffer and create image subroutines in the buffer. READ/WRITE commands are again used by the CPU to control the contents of the buffer.

The 2250 Model 4, which attaches to the IBM 1130 computing system, uses the main storage of the 1130 CPU to regenerate the display orders and data via the storage access channel. This main storage is available in many configurations up to 32,768 words (2 bytes/word) with a cycle time of 2.2 microseconds per word. Commands must still be sent to the 2250 because the registers that control and sequence the CPU storage addresses for display data

Figure 2 System configurations available with the IBM 2250



are physically located in the 2250 Model 4. Remote 2250 Model 4 installations can be provided by connecting the 1130 to a SYSTEM/360 computer over a 300-character-per-second voice grade telephone line or 5,100-character-per-second data transmission line.

Some of the highlights and capabilities of the various models of the 2250 are:

- A 12 × 12 inch working area on the screen
- 1024 × 1024 addressable points within this working area
- A 0.020 inch beam spot size
- 40 Hz (cycles per second) regeneration rate for flicker-free viewing with normal room lighting
- A character generator for Models 1 and 3 (set of 64 nonchangeable characters)
- A programmable character set using stroke tables in the 1130 system main storage for the Model 4
- Incremental and absolute end-point positioning for line drawing
- Two character sizes with space for 52 lines of 74 characters (basic) or 35 lines of 49 characters (large) in the working area
- Superscript and subscript capability in the Model 4

IBM 2250
capabilities

- Average time to display a character:
 - 11.9 microseconds—Models 1 and 3
 - 11.6 microseconds—Model 4 (with 2.2 μ s main storage option)
- Line drawing times in microseconds for:

	Model 1	Model 3	Model 4
Short line (3/16 inch)	9.3	7.3	8.9
Medium length line (2 inches)	22.4	21.4	24.0
Long line (8 inches)	67.8	67.8	72.8

- Subroutine linkage in Models 3 and 4
- Conditional branches and interrupts in Models 3 and 4

The input devices available for man-machine interaction on all 2250 models are the function keyboard, the light pen, and the alphanumeric keyboard. Figure 3 illustrates the basic functions of these input devices.

function
keyboard

A program function keyboard that can be moved contains 32 keys to which functions can be assigned according to the application program to be executed. Depressing a function key causes a direct CPU interrupt which in turn can be used to cause the execution of a particular subroutine. Plastic overlays containing written identification of the function of each key can be placed over the function keyboard. Notches cut into the overlays are sensed by the keyboard, and provide for use of up to 256 overlays by any application program. Each key also has an indicator light under it. These lights can be turned on and off under program control to provide cues to the user for subsequent action or key operation.

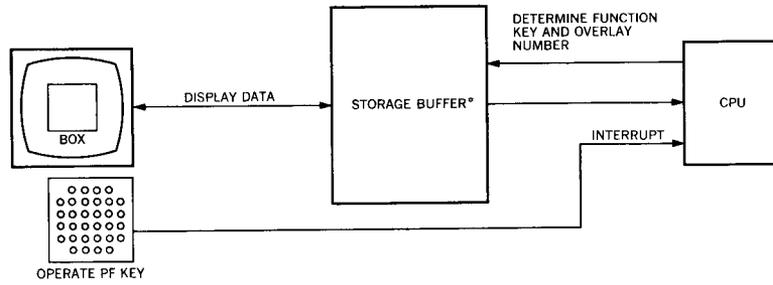
light
pen

The light pen, a light-sensitive device shaped like a pen, permits the user to select any item from the information displayed on the screen. The light from a point, line, or character being displayed triggers the light pen only during the instant in which the beam is actually under the pen. When the light pen detects light from the screen, regeneration of the display is usually stopped, and a CPU interrupt occurs. (The 2250 Models 3 and 4 provide deferred detects and interrupts as an alternate approach.) In this way, the buffer storage address of the data detected can be found, and, if applicable, the x - y position of the pen can be determined. The program may use this information to interpret the user's intent of the light-pen action.

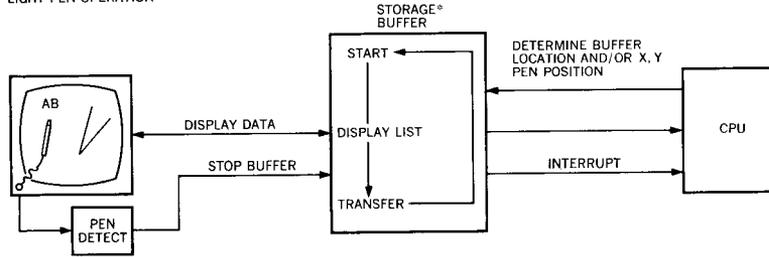
In many applications, new points or lines must be established on the screen where no data presently appears. To create these new points, light-pen tracking can be used. A symbol called a "tracking cross" (in simplest form it could be a small square or a cross made up of points) is displayed and is programmed to follow the movements of the light pen wherever it might wander within the working area of the screen. Orders are available for enabling and disabling the light-pen interrupt circuitry. Also, orders controlling the light pen allow pen "detects" to be deferred and tested by conditional branch orders instead of interrupts. CPU interrupts can be eliminated during tracking by the use of these orders. These are in addi-

Figure 3 Basic functions of IBM 2250 input devices

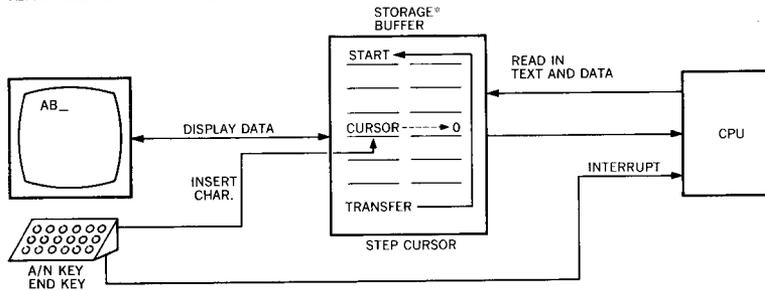
PROGRAMMED FUNCTION KEYBOARD OPERATION



LIGHT PEN OPERATION



ALPHANUMERIC KEYBOARD OPERATION



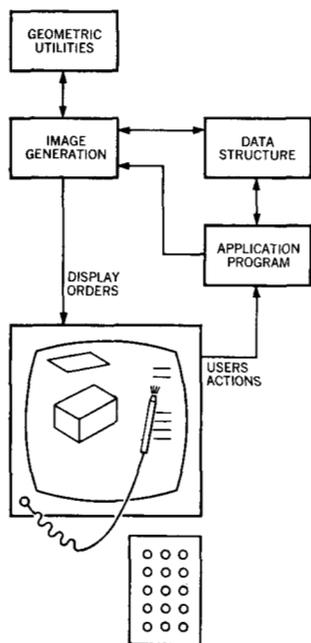
*STORAGE BUFFER IS PART OF 1130 STORAGE FOR 2250 MODEL 4

tion to facilities provided by CPU programming for disabling, enabling, and queuing of interrupts.

In the Models 1 and 3, the alphanumeric keyboard can be used to insert numeric data and text directly into the display buffer without disturbing the CPU. The Model 4 requires CPU interrupts for keyboard operation, since it uses the main storage of the 1130 system as a buffer. A cursor displayed on the screen indicates where the next character entered from the keyboard will be displayed. The cursor automatically moves to the next character position as each character is entered. Complete text messages and an entire

alphanumeric keyboard

Figure 4 Interactive display systems



panel of numeric parameters can be entered into the buffer of the Models 1 and 3 before the END key is hit, causing the first CPU interrupt, thus allowing a READ buffer command to bring the information into the CPU. These cursor functions are provided by programming in the 2250 Model 4.

Image generation

We now turn from a description of machine functions to the programming requirements for creating display images.

One method of representing an interactive display system is as a feedback loop in which the display console serves as a means of input to the application program. A diagram of such a feedback loop is shown in Figure 4. In this diagram, the application program communicates with a number of functional modules which are referred to as "packages." The application program interfaces with a data structure package, an image-generation package, and a geometric utility package.

The data structure package has access to the canonical forms (standardized, mathematical-algorithmic descriptions) of the geometric objects which appear in the display. In addition, it has access to alphanumeric information, and information describing the attributes and interrelationships of the different portions of the display which may be required by the application program.

The application program receives this data and passes it on to the image-generation package. If geometric transformations are required, the geometric utility routines manipulate the data. After a suitable transformation to obtain the desired view or projection, the image-generation package creates the display orders for the required picture.

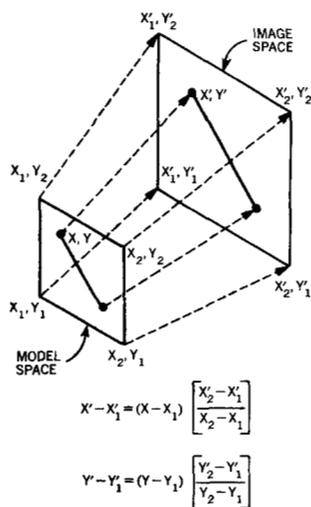
Once the picture has been displayed on the console screen, the user can interact. He takes certain actions using either the function keyboard, the alphanumeric keyboard, or the light pen, so as to invoke program functions, and these program functions generally modify the picture, thereby completing the feedback loop.

The image-generation operations can be further broken down into two major categories: (1) creation of the image and (2) control of the image. The discussion of image creation is not limited to interactive CRT displays, but is equally applicable to drafting machines and plotting devices.

To create a display, the image generator must map a planar figure from model space to a defined image space. Figure 5 shows a finite area of a two-dimensional model space mapped onto the 2250 screen (image space). The mapping in the illustration transforms two-dimensional model coordinates into display device coordinates and is a straightforward proportional mapping. A discussion of other projective transformations can be found in the paper by Ahuja and Coons which is included in this issue.

To create the image, the program collects all of the elements from the data base that are contained within the limits of the

Figure 5 Mapping



display. A three-dimensional model must first be mapped onto a plane to create a two-dimensional model. The two-dimensional model is then mapped to display device coordinates for creation of the image.

Images that fall partially off the image space must be "scissored" by the image-generation program. Scissoring is a term applied to the truncation of a vector which crosses the boundary of the image space. Figure 6 illustrates scissoring of a vector.

Since the 2250 can generate only points, straight lines, and alphanumeric characters, all pictures must be composed of these "primitive" elements. Proper use of these primitives must be relied on to achieve a desired pictorial effect. As an illustration, consider the display of an arc as shown in Figure 7. The image of the arc is usually composed of a series of connected straight lines, or vectors, where the length of these short chords is chosen such that the distance T (the deviation from the true curve) is within tolerable limits for viewing. Another method of displaying the same arc is with a series of closely spaced points. However, fewer vectors are needed to present a smooth curve than would be required if it were composed of points. Since each incremental vector requires the same amount of information as does each point, vector displays require less buffer space. To achieve an image where the distance T is constant is relatively simple when an arc of constant curvature is drawn. For arcs where the curvature is not constant, as in conic or spline curves, the tolerance T must be evaluated for a local radius of curvature and the chord length adjusted to attain an apparently smooth curve for viewing.

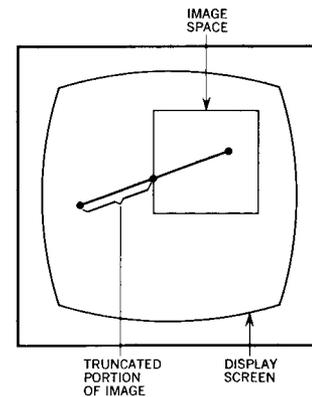
Certain computational methods, such as those of projective geometry, can produce a fairly uniform visual tolerance as a more or less intrinsic consequence of their use. These methods, described in the following paper by Ahuja and Coons, are to be favored for their computational simplicity and uniformity.

Alphanumerics on the display are generally produced in one of two ways: either the terminal display device incorporates a character generator feature, thereby permitting machine generation of characters, or as an alternate, the device can produce alphanumeric characters by means of a stroke table. The stroke table breaks alphanumeric characters down into a set of short vectors. A suitable transformation upon this vector set creates alphanumeric characters of proper placement, orientation, and scale.

Control of the interactive display is accomplished by utilizing the display console input features and program techniques. By keying in or typing in different commands to an application program, the user can initiate various program actions, and thereby exercise on-line control over the application program. In addition, the application program can assign correlation numbers for image control. Correlation numbers are program-assigned to portions of the displayed image and serve as identifiers or names. When a light-pen detection occurs, the application program has access to the assigned correlation number of the detected element.

use of
elements

Figure 6 Scissoring



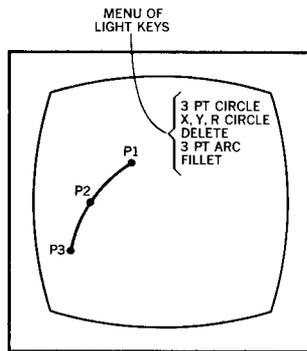
alphanumeric
display

control of
display

Figure 7 Display of an arc



Figure 8 Light keys



Correlation numbers combined with the use of the light pen provide the user with a great deal of flexibility in the design of his control functions. This combination permits the use of displayed "light keys." Light keys appear like a list, or "menu," of "commands" on the display screen. They can be placed anywhere on the screen. A selection with the light pen on any light key will activate a particular function at that point in the execution of the program.

In Figure 8, a menu of functions using light keys is displayed in the upper right corner of the CRT screen. These functions are typically useful in drafting applications for the construction of circles and arcs. As an example, the operator could obtain a light-pen detect on the menu item 3 PT ARC and could then detect on three previously displayed (noncolinear) points on the screen. This procedure would result in the computation and display of an arc through the three points. If, instead, the operator detected with the light pen on the item 3 PT CIRCLE (and the three points), the program would generate a full circle through the points (up to the limits of the CRT image space). To delete a displayed circle or arc, he could detect serially on the menu item DELETE and on the arc.

Use of the menu item X, Y, R CIRCLE generates a full circle, but the parameters (x coordinate and y coordinate of the center point) and the radius parameter are keyed in by using the alphanumeric keyboard. The menu item FILLET is used to compute and display an arc that is tangent to two detected line segments. The radius of the arc is entered into the system by using the alphanumeric keyboard. In addition, the application program could display tutorial or diagnostic comments at any desired location on the display screen.

Concluding remarks

The ability to display complex drawings, pictures, curves, and graphs along with the closely interactive input devices (particularly the flexibility available with the light pen) makes a graphic display console such as the IBM 2250 a practical interface for blending man's creative talents with the computational power of a computer required for problem-solving. The varied models and system configurations of the 2250 and similar devices serve as a solution to many graphic interaction problems. However, they do not constitute a complete solution. There is always room for improvement and a desire to extend the state-of-the-art with more sophisticated machine functions.

The graphic capability of the display could be improved by tone control¹³ and color rendering in varied chroma, saturation, and hue.¹⁴ Provision could also be made for semitransparent display with control over the degree of transparency. These characteristics could then enable phantom drawing or overlay drawing, which is possible with the RAND tablet.¹² It would also be useful to develop time as a control parameter to synchronize display elements with events such as music, industrial processes, etc. For example,

cartoon animation would require time control. Time control would also be useful for interaction where the display serves as an interface with a time-dependent process. An example of such an interaction is the real-time display of a chemical reactive process.

The creation and error checking of control tapes for numerically controlled machine tools can be simplified and greatly improved by using the display console to specify the cutter paths.¹⁵ Similarly, the creation of color printing masks for producing textile prints is made much easier through the use of display consoles.¹⁶ Thus, a graphic display console is a most useful tool in the design and analysis stages of many products and processes.

ACKNOWLEDGMENT

The authors gratefully acknowledge the contributions of J. L. Cleveland to the discussion on the IBM 2250.

CITED REFERENCES

1. J. D. Aron, "Real-time systems in perspective," *IBM Systems Journal* **6**, No. 1, 49-66 (1967).
2. I. E. Sutherland, "SKETCHPAD: A man-machine graphical communication system," *AFIPS Conference Proceedings, Spring Joint Computer Conference* **23**, 329-346 (1963).
3. T. E. Johnson, "SKETCHPAD III—3D graphical communication with a digital computer," *AFIPS Conference Proceedings, Spring Joint Computer Conference* **23**, 347-353 (1963).
4. G. S. Devere, B. Hargreaves, and D. M. Walker, "The DAC-1 System," *Datamation* **12**, No. 6, 37-47 (June 1966).
5. F. Gagliano, H. W. Thombs, and R. E. Cornish, "A conversational display capability," in this issue.
6. H. B. Baskin and S. P. Morse, "A multilevel modeling structure for interactive graphic design," in this issue.
7. J. D. Jacobsen, "Geometric relationships for retrieval of geographic information," in this issue.
8. R. C. Wood and P. Hendren, "A flexible computer graphic system for architectural design," *Information Display* **5**, No. 2, 35-40 (1968).
9. A. Appel, "The notion of quantitative invisibility and the machine rendering of solids," *Proceedings of the 22nd National Conference of the Association for Computing Machinery P-67*, 387-393 (1967).
10. F. S. Greatorex, Jr. and D. Cohen, "Producing dynamic perspective views for vehicle simulation," *Data Processing Magazine*, 26-31 (April 1968).
11. L. G. Roberts, "The Lincoln Wand," *AFIPS Conference Proceedings, Fall Joint Computer Conference* **29**, 223-227 (1966).
12. M. R. Davis and T. O. Ellis, "The RAND tablet: A man-machine communication device," *AFIPS Conference Proceedings, Fall Joint Computer Conference* **26**, 325-331 (1964).
13. C. Wylie, G. Romney, and D. Evans, "Half-tone drawings by computer," *AFIPS Conference Proceedings, Fall Joint Computer Conference* **31**, 49-58 (1967).
14. A. Appel, "Modeling in three dimensions," in this issue.
15. T. P. Dankowski, "Computer graphics and NC—a feasibility study," *Numerical Control Society Proceedings, Fourth Annual Meeting and Technical Conference*, 50-65 (1967).
16. J. R. Lourie and J. J. Lorenzo, "Textile graphics applied to textile printing," *AFIPS Conference Proceedings, Fall Joint Computer Conference* **31**, 33-40 (1967).