The IBM 3277 Graphics Attachment is an interactive computer graphics workstation using a dual-screen concept. A storage display monitor is attached to an IBM 3277 Display Station, the combination providing low-cost, moderate-performance interactive graphics. This paper describes the architecture of the Graphics Attachment, both functional structure and rationale. Nonarchitectural characteristics are also considered.

Architecture of the IBM 3277 Graphics Attachment

by D. F. McManigal and D. A. Stevenson

Interactive computer graphics is one of the most rapidly growing fields within the data processing industry. This growth stems from the very large productivity gains that computer graphics makes possible and also from the decreasing cost of graphics workstations.

A minimal interactive graphics workstation comprises a device for displaying pictorial data and a device for accepting pictorial input. Usually the display device is a cathode ray tube (CRT) and the input device is a pointing mechanism such as a light pen, a digitizing tablet, or cursor-positioning thumbwheels.

Most interactive graphics workstations also include a device for displaying alphanumeric output data and for accepting alphanumeric input data. Usually, the alphanumeric display device and the graphics data display device are one and the same. The alphanumeric input device is usually a keyboard.

Graphics displays of many kinds are used, but three types of cathode ray tube displays lead the field:

Refreshed raster-scan displays include a means of translating binary input values into output illumination intensity levels for every displayable point on the screen. Each displayable point shown in Figure 1 is scanned and illuminated to the specified intensity value once during each refresh cycle. Typical refresh rates range from 25 to 60 cycles per second.

Figure 1 Concept of refreshed raster-scan display

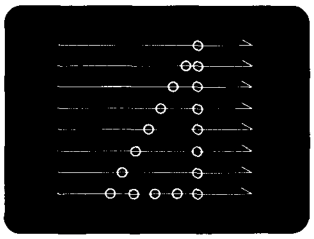
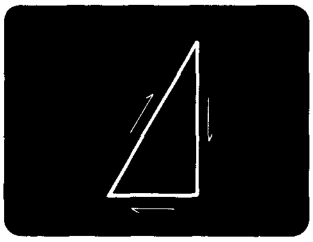


Figure 2 Concept of refreshed directed-beam display



Refreshed directed-beam displays present graphics data in tip-to-tail vector form. As shown in Figure 2, each vector is a straight line connecting two endpoints and is regenerated during each refresh cycle. Unlike the raster-scan display, directed-beam displays scan only the paths between vector endpoints.

Storage tube displays use cathode ray tubes that incorporate a means of storing displayed data and causing them to remain visible without refreshing once written. The most common device of this class is the direct-view storage tube (DVST), which derives its name from the fact that the phosphor layer that produces the viewable image is also the storage medium. The DVST uses the directed-beam technique and is capable of displaying refreshed data in addition to stored data.

The IBM 3277 Graphics Attachment[1] (Figure 3) consists of circuitry added to the 3277 Display Station to provide all analog and digital signals required for displaying graphics data on an attached storage display monitor and for accepting and transmitting graphics input data. The 3277 is essentially unchanged; it continues to perform its normal alphanumeric display and input functions. Graphics data and orders are encoded as 3277 characters and embedded in ordinary 3277 data streams. This attribute permits the Graphics Attachment to be used without special operating systems support, because it appears to the host system exactly as an ordinary 3277 display station. All graphics programming support is contained in application programs or in a library of application subroutines.

The Graphics Attachment includes a display-refreshing capability, permitting both stored and refreshed data to be displayed simultaneously on the same screen. This refreshed-display facility is extremely valuable in providing greater interactivity than can be achieved by displaying only stored data.

The Graphics Attachment also includes two serial communication ports, one for input and one for output devices. By means of these ports, various auxiliary devices, such as a digitizing tablet and an output plotter, may be attached to the 3277, as shown in Figure 4.

Background

Interactive computer graphics is not a new field. In the early 1950s, it was a research subject. By the late 1950s, it had become a tool of large, complex military systems, such as the air defense system known as SAGE. In the early 1960s, IBM undertook a joint development effort with the General Motors Research Laboratories. The result of that effort was the design and development of a

Figure 3 Dual-screen concept of the IBM 3277 Graphics Attachment

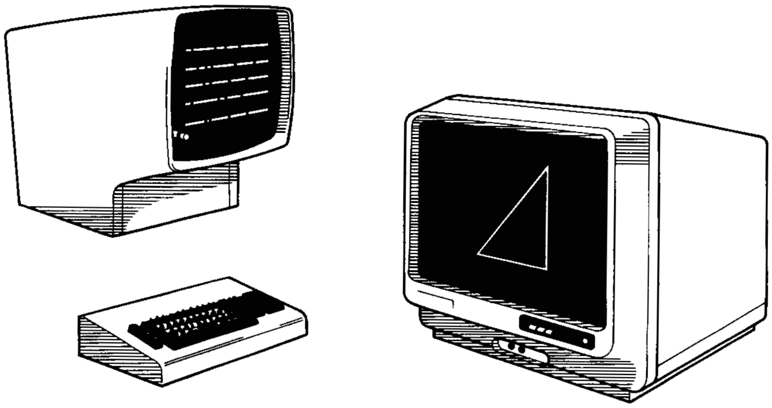
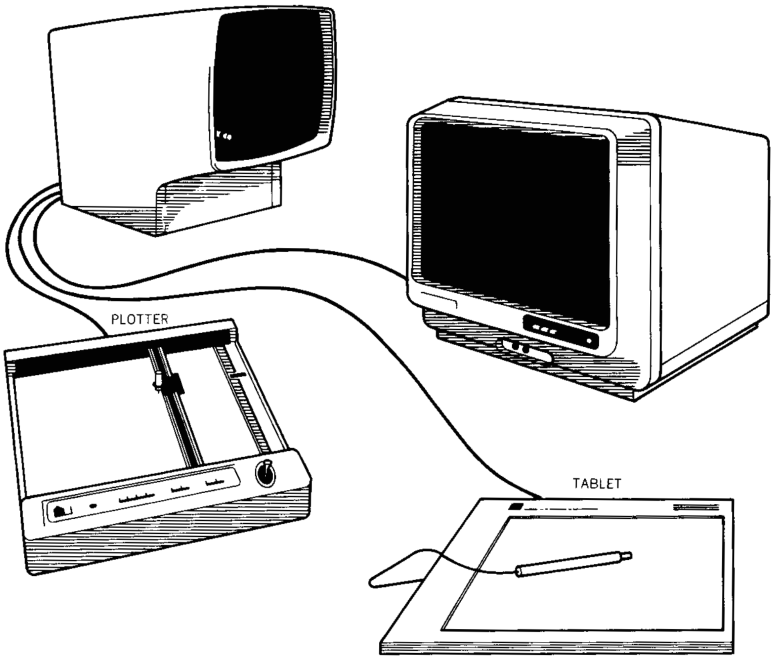


Figure 4 Input and output port concept with example I/O devices



special image-processing system.[2] This activity led to the development of the IBM 2250 Display System, which was announced in 1964.

The 2250 gained wide acceptance as an effective interactive computer graphics workstation. Among its well-known applications are the one at the General Motors Research Laboratories discussed in Reference 2 and the computer-aided design system of the Lockheed Corporation known as CADAM®.

The 2250 provided the basic graphic functions—the ability to display vectors and characters and the ability to handle graphics input from the light pen—and it performed these functions at high speed. Therefore, it was able to support high rates of interaction, on the order of several transactions per second.

Because of the substantial hardware costs involved, and because the interface to the 2250 was unique, graphics applications using that system had to be well-defined and justified before equipment could be ordered and implementation begun. Only the extremely high productivity increases obtainable through interactive graphics could maintain interest in the field during this period. Widespread use of computer graphics had to await less expensive facilities.

In the late 1960s and early 1970s, low-cost graphics terminals were developed. Some of these terminals depended on the reduced cost of the new integrated circuit technologies to permit the addition of processing power to the graphics terminal itself. This allowed relatively low-cost terminals to perform advanced functions, including emulation of the IBM 2250 and local manipulation of displayed data without host system intervention. Such terminals offered the advantage of reduced hardware cost, but they did not reduce system complexity or programming costs.

In the late 1960s, other low-cost graphics terminals were designed that used storage tube displays. These terminals operate almost entirely in stored mode, using refreshed mode only to display a graphics cursor as an indicator of graphics input position. They have three advantages over other graphics workstations:

- Low cost.
- Emulation of widely used alphanumeric terminals.
- Ease of use.

Such a terminal can be added to an existing system and used immediately as an alphanumeric terminal. Without the need to manage a graphics refresh buffer, graphics input and display functions may be programmed as they occur, simplifying graphics programming and permitting graphics applications to be developed in an incremental fashion. This incremental development capability has proved to be extremely valuable in expanding the field of interactive graphics, because it allows new applications to be explored before they are fully defined.

The dual-screen workstation

The architectural approach used in the 3277 Graphics Attachment results in a graphics workstation comprising two display screens, an alphanumeric keyboard, and a graphics input joystick. The dual-screen workstation is a departure from the conventional approach, which uses a single dual-purpose screen.

The availability of two screens is one of the major advantages of the 3277 Graphics Attachment. Most graphics applications involve both graphics and alphanumeric data. At the minimum, it is necessary to issue system commands to initiate a working session. Beyond this, most graphics applications require the entry and subsequent editing of textual data—drawings require annotation, flowcharts require step descriptions, maps require street names, etc.

The 3277 is a well-supported and widely accepted alphanumeric input/output terminal. Therefore, it is a good choice for alphanumeric functions, whether or not those functions are part of a graphics application. Additionally, direct-view storage tube (DVST) display devices have been shown to be very useful in moderate-performance graphics applications, combining low cost with high data density and ease of programming. So it is desirable to use each display device for the functions in which it excels, degrading neither for the sake of the other. This is the dual-screen workstation concept.

example of an application

The use of two display screens in a single application is illustrated in Figures 5 and 6. The application in this example is the interactive generation of a business-related graph using Programming RPQ P09013.[3–5] Figure 5 shows the 3277 display, which is used for editing. Figure 6 shows the resulting graphics display. Developing a graph of the desired form often involves several iterations of the editing and display processes. It is useful during editing to retain the previous graphics display frame for reference, and this is easily accomplished using the dual-screen workstation.

Architecture of the dual-screen graphics system

The IBM 3277 Graphics Attachment is a field-installable extension of the IBM 3277 Display Station that permits the attachment to the 3277 of up to three additional graphics devices:

- One directed-beam storage display monitor, such as the Tektronix Model 618.
- One input device, such as a digitizing tablet, utilizing 7-bit ASCII code and an RS232-C serial interface.
- One output device, such as a vector plotter, also utilizing 7-bit ASCII code and an RS232-C serial interface.

Figure 5 Example of 3277 display as used for editing

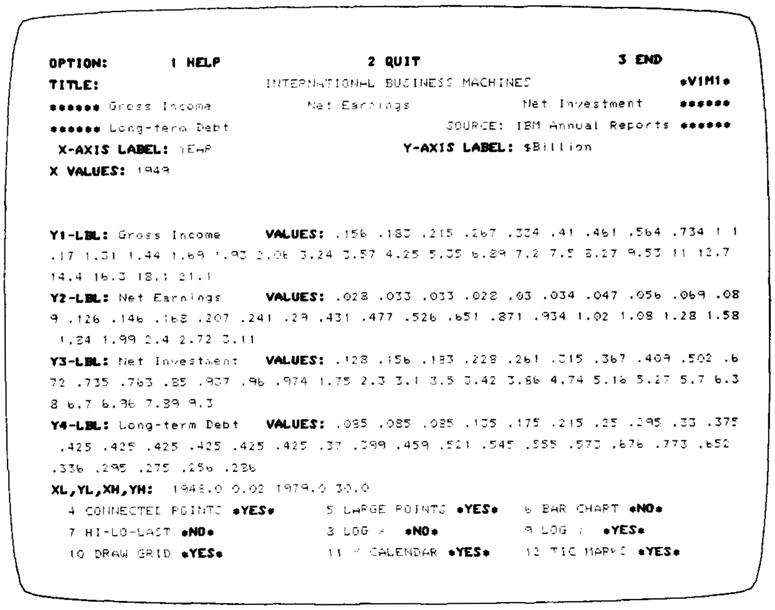
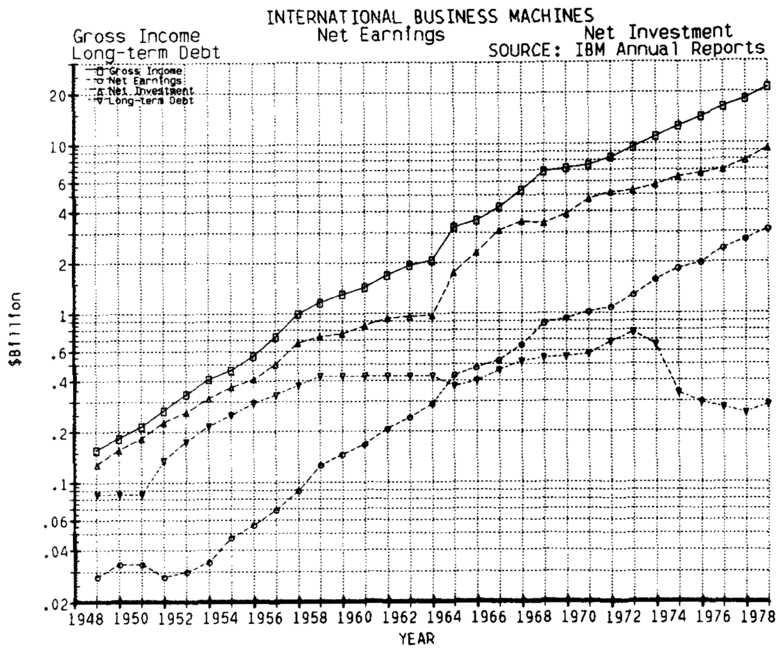


Figure 6 Example 3277 Graphics Attachment output display



The Graphics Attachment includes a graphics input facility that allows an application program to display a graphical object on the attached storage display monitor in refreshed mode, and then allows the terminal user to move the object to a desired screen position and transmit this position to the application program. This graphics input function provides the essential difference between *interactive graphics* and *presentation graphics*, the latter being the noninteractive display of graphics data.

All elements of this architecture are intended to maximize performance within a cost objective. A refreshed CRT display station would provide higher performance than a storage display station, but it would cost more. Specific direct interfaces for tablet and plotter could provide higher performance than the RS232-C ports of this architecture, but they would cost more. The CRT display interface could be omitted at a cost saving, but a workstation using only tablet and plotter could not be characterized as "medium-performance."

With these considerations in mind, a workstation architecture consisting of the following components was defined:

- Graphics interface and buffer.
- Graphics command decoder.
- Vector generator.
- Programmable pattern generator.
- Graphics input controls.
- RS232-C interface.

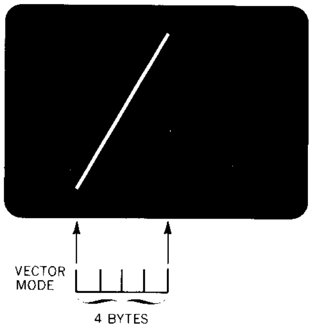
Each component of the architecture works with other components, yet each is essentially independent of the others. A component may not be needed in some applications, but its exclusion would limit the broad applicability of the graphics workstation. Thus, this architecture represents, in effect, a kit of tools for handling most low- to medium-performance graphics applications, rather than a solution tailored to any one problem set.

Graphics interface and buffer

The graphics interface and buffer capability performs three functions:

- It monitors all data sent from the host system to the 3277 as the data are being written into the 3277's alphanumeric display buffer. Upon recognition of a graphics escape character, subsequent output data are stored in the graphics buffer.
- It routes graphics input data from the Graphics Attachment into the 3277's alphanumeric display buffer by simulating keyboard input of alphanumeric characters.
- It retrieves graphics output data from the graphics buffer for execution by the Graphics Attachment. Execution may occur once (in the normal output mode), or it may be repeated to display refreshed or blinked objects.

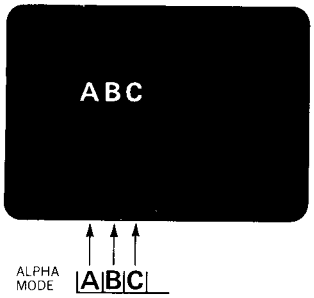
Figure 7 Concept of the vector data mode



Execution of the contents of the graphics buffer begins upon completion of buffer loading and terminates upon completion of one cycle in normal (nonrefreshed) output mode. Alternatively, execution terminates upon commencement of another graphics output operation, or upon the completion of a graphics input operation.

In essence, the graphics interface and buffer capability is a switch that is activated by the graphics escape character and deactivated by another character that signifies the end of the graphics buffer load. When the switch is activated, data transmitted from the host system to the 3277 are diverted to the graphics buffer. When it is deactivated, the 3277 performs its normal functions, and the contents of the graphics buffer are made available to the command decoder. The primary purpose of the graphics buffer is to synchronize the high burst speed of the alphanumeric output to the 3277 with the variable speed of interpreting and executing graphics commands.

Figure 8 Concept of the alpha data mode

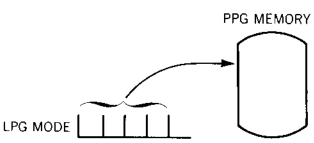


Graphics command decoder

There is a graphics command decoder that accepts data from the graphics buffer, interprets it, and generates control signals for conditioning other components to execute the indicated graphics functions. The command decoder also provides graphics modes that effectively extend the range of available data codes by establishing different meanings for some codes, depending upon the current mode. Other modes control output signals, such as stored or refreshed display, or data routing, such as graphics display or RS232-C output port data selection. Other control codes provide direct control signals to reset the Graphics Attachment, erase the graphics display screen, or trigger an audible alarm signal.

Three data modes are provided by the graphics command decoder: vector mode, alpha mode, and load pattern generator mode. The vector mode, which is illustrated in Figure 7, causes groups of two or four characters to be interpreted as X-Y vector endpoint values. The alpha mode, shown in Figure 8, causes each output character to select a particular set of vectors that had been previously loaded into the programmable pattern generator. The programmable pattern generator is initialized in the load pattern generator (LPG) mode shown in Figure 9.

Figure 9 Concept of the load program generator mode and programmable pattern generator memory



The primary order codes currently recognized by the graphics command decoder represent a subset of possible graphics workstation functions. Unused primary codes are available for the implementation of other functions, and an escape code is provided to support a virtually unlimited extension of the presently defined function set. This escape code alters the meaning of one or more characters that follow it, thereby permitting new graphics commands to be defined without adding new primary command codes.

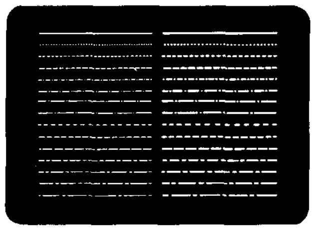
Vector generator

The vector generator is essentially a mechanism that transforms a pair of vector endpoints into a straight line between the endpoints. The vector generator retains the last-used vector endpoint and accepts as input a new vector endpoint. Upon receipt of input, the vector generator outputs all intermediate X-Y addresses to define the closest possible approximation to a straight line between the two endpoints. Because the vector address space is defined in the range 0–4095 in each axis, vectors are very close approximations of straight lines. In a workstation employing a 19-inch diagonal display screen with a nominal 0.3-mm spot size, the maximum deviation amounts to about one third of a line width.

There are three vector formats. Absolute vectors are specified by their absolute endpoint addresses, expressed as four bytes of data. Long relative vectors are specified by the displacement in each axis from the vector origin to the vector end. Maximum displacement in each axis is 2047 units. Each vector requires four bytes of graphics data. Short relative vectors are like long relative vectors, but the maximum displacement is 127 units, and each vector requires only two bytes of graphics data.

In addition to the three display modes provided by the graphics interface and buffer (stored, refreshed, or blinked), vectors may be displayed in any of fifteen basic line types (solid, dotted, dashed, dot-dashed, etc.), and stored or blinked vectors may be displayed in either of two line widths, thin (normal) or thick (defocused). Figure 10 illustrates thirty distinct line types (the fifteen basic line types, previously mentioned, displayed as thin and thick) that may be displayed in stored or blinked mode. The same line types are available in refreshed mode, but refreshed vectors may not be displayed as thick lines.

Figure 10 Samples of the fifteen line types in two line weights available on the 3277 Graphics Attachment



Each of these vector characteristics has its uses. Of course, most data are displayed in the normal mode, i.e., stored, solid, thin-line vectors. This is the default mode established when vector mode is entered without selecting other mode options. The use of line types other than solid is of value when objects of different types are identified to the terminal user. For example, dimension lines in a drawing can be shown as dotted lines, or the various curves in a multiplot graph may be identified by using a different line type for each.

The primary usefulness of relative vectors is to define graphics objects that are to be displayed in more than one location, and the

Figure 11 Characters as displayed either (a) as a matrix of dots or (b) as a series of vectors

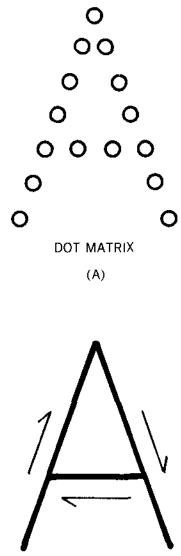
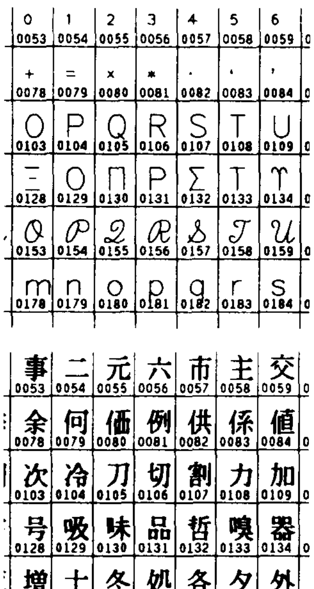


Figure 12 Samples of patterns available on the 3277 Graphics Attachment



primary usefulness of the refreshed mode is to display objects that are to be moved from one location to another in a graphics input operation.

Stored vectors may be highlighted in three ways: by thickening only, blinking only, or blinking and thickening them. Thick vectors are more noticeable to the user than normal vectors, and blinking provides additional highlighting.

In addition to the various forms of visible vectors, invisible vectors are used to permit the positioning of the display beam at the beginning of a series of vectors that define an object. Invisible vectors are generated by MOVE operations, whereas visible vectors are generated by DRAW operations. MOVE operations need not occur as straight lines between the two endpoints; rather, they may position the display beam at the fastest possible rate to a new endpoint, regardless of the path the beam may take.

The scaling of relative vectors by integer multipliers in the range 0–63 is another basic function of the vector generator. The default scale value is 1, but any value in the range 0–63 may be set by an order from the host system. When any relative vector order is received by the vector generator, the specified X-Y displacements are multiplied by the current scale factor. A zero scale factor reduces all relative vectors to points.

Programmable pattern generator

Because most graphics applications use alphanumeric data, it is usual for a graphics workstation to include a character generator. Characters are displayed in one of two ways: as a matrix of dots (Figure 11a), or as a series of continuous vectors (Figure 11b). The dot-matrix approach is always used for raster-scan displays because it is easily and directly implemented, as illustrated in Figure 1.

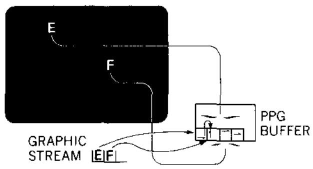
The continuous-vector approach to character generation, which is often referred to as "stroking," is the preferred technique for directed-beam displays. Whether stored or refreshed, stroking produces more legible characters at a higher speed than does the dot-matrix approach. Stroking is also more flexible than matrix techniques. Stroked characters may be in any shape, size, aspect ratio, or position because all such attributes may be included in the character definition. Such a generator goes beyond character generation and becomes pattern generation, where the term "pattern" denotes any arbitrary shape. The pattern generator has the additional attribute of programmability, meaning that it may be loaded with a pattern set transmitted to it by an application program residing in the host system. The workstation component that provides these capabilities is termed the programmable pattern generator (PPG).

Figure 12 shows samples of pattern sets taken directly from the screen of a graphics display monitor. These full-size characters were obtained from general-purpose font files[6] and displayed as patterns of vectors using the 3277 Graphics Attachment. Although only a few characters from each font are shown here, each font contains 375 characters.

alpha mode

The programmable pattern generator (PPG) comprises a pattern memory and pattern storage and retrieval controls. Patterns are stored as series of short relative vectors and branch orders. A branch table is loaded into the beginning of the pattern memory. Each character code received in "alpha" mode is used as an index into this table, and pattern retrieval begins at the index address. The index normally addresses a branch order that directs the PPG to the first vector order of the pattern. Subsequent relative vector orders are executed sequentially until an order is encountered that is flagged as the last of the pattern. Other branch orders may occur within a pattern definition to permit common sequences of vectors to be shared by several patterns as shown in Figure 13. Each vector order within a pattern definition may also contain a RESET flag that provides a rough equivalent of the RETURN function of many alphanumeric terminals, except that the return may be in either axis.

Figure 13 Concept of the programmable pattern generator



Because all vector orders extracted from the PPG memory are presented to the vector generator as relative vectors, they are scaled according to the current scale factor before they are displayed. Also, patterns are displayed according to the current display mode, such as stored/refreshed/blinked or normal/defocused, except that they are always displayed as a solid line type, regardless of the current vector line type value. (A dotted alphabet, for example, may have a use, but it must surely be an obscure application.)

In addition to displaying characters of virtually any font, the PPG is also useful as a graphics subroutine facility. Frequently used graphics shapes, such as engineering symbols, business logos, etc., may also be loaded into the PPG along with a standard font, and then be treated as simple characters.

Graphics input controls

The key element of the graphics input controls is the *tracking register*. This register may be set to any X-Y position value by orders from the host system and altered by the terminal user by means of a graphics input device. The following four primary graphics orders affect the tracking register:

- *Initialize Tracking Register* (ITR) causes the tracking register to be set to a specified absolute X-Y position.
- *Move to Tracking Register* (MTR) causes the graphics display beam to move to the current tracking register position.
- *Draw to Tracking Register* (DTR) draws a vector from the last previous vector endpoint to the current tracking register position.
- *Read Tracking Register* (RTE, RTN) activates the graphics input controls. Thus, striking a key on the 3277 keyboard causes the current tracking register value to be encoded as a four-byte EBCDIC string and placed in the 3277 data buffer following the byte representing the struck key. RTE causes an ENTER code to be appended to the four graphics bytes, resulting in immediate input transmission to the host system. The RTN does not generate the ENTER code. Thus, when RTN is used, the terminal operator must strike the ENTER key on the 3277 keyboard after entering the last point of a graphics input sequence to cause the entire sequence to be transmitted to the host system. In summary, RTE activates the input of a single point, and RTN activates the input of one or more points.

The value of the tracking register may be changed during a refreshed display operation by means of a joystick, which is a spring-loaded lever that takes its name from the aileron and elevator control stick of an airplane. The joystick may be manually deflected in any X-Y direction. The direction and degree of deflection determine the direction and speed of change in the tracking register position.

A typical graphics input operation includes an ITR order to position a graphics object at the desired default position, an MTR order to make the tracking register position the current graphics reference position, a series of relative vector orders to display an object at the reference position, and an RTE order to begin the graphics input transmission. The terminal user then sees an object appear on the screen at a designated position, moves the object using the joystick, and signals the final position to the host system by pressing a key on the 3277 keyboard. The user's choice of key may be interpreted by the application program as an order, thereby making each graphics input sequence a complete command containing both operation and position.

The graphics input device need not be a joystick, but may be a tablet, tracking ball, "mouse," or thumbwheels, all of which are used in one form or another in current graphics workstations. The device used to control the tracking register is not important to the architecture; however, the tracking register function is important, because it serves as an effective and generalized graphics *locator* device.

Graphics locators differ from graphics *selectors* or pick devices,[7] such as the light pen, because they provide a graphics position without regard for what may be displayed there. A selector, however, can indicate only the chosen displayed item.

It is worthwhile to note that a major function of selectors is that of picking menu options. Although this is not strictly a graphics function, it is supported quite well by the 3277 light pen option and is in no way restricted by the Graphics Attachment.

The RS232-C interface

The RS232-C interface in the 3277 Graphics Attachment permits the attachment of a limited selection of graphics and nongraphics devices that are not specifically supported by the architecture. The RS232-C interface comprises one input port and one output port, each of which may operate at instantaneous speeds up to 19.2 kilobaud. The output port is also capable of limited transmission in the opposite direction, such as a few bytes of status information received from the output device. The input speed is limited to an average data rate of 300 baud or less because characters are received at 3277 keyboard rates. A 16-character buffer permits the instantaneous rate of 19.2 kilobaud for short bursts. This is not meant to be a general-purpose communication facility, but one that fills a specific need for certain types of devices.

Each RS232-C port is activated by a specific graphics order. The data byte that follows each graphics order establishes the port characteristics, such as baud rate and, for input, the terminator character codes. Once activated, the output port transfers all subsequent data contained in the indicated graphics buffer to the output port at the specified baud rate.

When the input port is activated, it transmits input data to the 3277 buffer at the current 3277 cursor position until the terminal operator strikes the ENTER key on the 3277 keyboard or until the specified terminator code is received in the AUTO-ENTER mode.

Concluding remarks

This paper has described an innovative approach to interactive computer graphics for low-cost, moderate-performance applications. The system uses a widely accepted alphanumeric terminal as a base for an interactive graphics workstation. Graphics orders and data are embedded in streams of conventional alphanumeric data. User-friendly, evolutionary development of graphics applications is thus promoted. This is made possible in part by the use of display technologies that are well-suited to the specific display task at hand. The dual-screen workstation is highly effective, largely because each display device is used in its most effective mode. Architecturally, the system has been designed around a subset of known graphics techniques that is complete enough to satisfy a large number of users in many fields. The IBM 3277 Graphics Attachment embodies these concepts in a practical workstation that we hope will satisfy many longstanding requirements.

CITED REFERENCES AND NOTE

1. *IBM 3277 Display Station, Graphics Attachment RPQ 7H0284, Custom Feature Description*, IBM Systems Library, order number GA33-3039, available through the local IBM branch office.
2. E. J. Jacks, "A laboratory for the study of graphical man-machine communications," *1964 Fall Joint Computer Conference Proceedings* **26,** Part 1, 343–350 (1964).
3. *IBM 3277 Graphics Attachment Support (Programming RPQ P09013), General Information Manual*, IBM Systems Library, order number GH20-2146, available through the local IBM branch office.
4. *IBM 3277 Graphics Attachment Support (Programming RPQ P09013), Program Reference and Operations Manual*, IBM Systems Library, order number GH20-2146, available through the local IBM branch office.
5. *IBM 3277 Graphics Attachment Support (Programming RPQ P09013), Logic Manual*, IBM Systems Library, order number GH20-2146, available through the local IBM branch office.
6. The character fonts used here are derived from NBS Magnetic Tape No. 12, which is described in NBS Special Publication 424 (available from the National Technical Information Service, United States Department of Commerce, Springfield, VA 22161, as Report No. NBS SP-424).
7. D. L. Weller, E. D. Carlson, G. M. Giddings, F. P. Palermo, R. Williams, and S. N. Zilles, "Software architecture for graphics interaction," *IBM Systems Journal* **19,** No. 3, 314–330 (1980, this issue).

The authors are located at the IBM Data Systems Division Laboratory, Box 390, Poughkeepsie, NY 12602.

Reprint Order No. G321-5128.