Raster-scan tube adds to flexibility and lower cost of graphic terminal

An ingenious scan-conversion algorithm recasts computer data into a form that drives a raster-scan display in real time; interaction of the display with the computer is also optimized

by Daniel E. Thornhill* and Thomas B. Cheek,** *Adage, Inc., Boston, Mass.*

☐ The graphic display of computer output, particularly in the case of complicated images, is still a costly proposition. On the other hand, TV-like raster-scan tubes and their supporting circuitry are inexpensive. The chief obstacle to their use in computer terminals, however, has been the complexity of converting computer data into the raster-scan signal format.

That obstacle has been overcome by a new approach to graphic terminal design, in which a real-time scan-conversion algorithm keeps to a minimum the amount of local memory the display device needs and also optimizes the device's interaction with the computer. The display device is a raster-scan monitor, which promises great flexibility in system design.

Just as alphanumeric symbols are the normal language elements for text presentation, so pictures of graphic entities are the language elements of mathematics, mechanics, architecture, and many other fields. While some of these elements might be presented as special symbols by an alphanumeric-only generator, or constructed from a small repertoire of "vector symbols" (often called limited graphics), others are too complex in shape and positioning to be so handled. General-purpose graphics—in which a vector can be drawn between any two points in the display field—are the only complete answer.

Several different approaches have been used in building graphic displays (see "The other types of graphic display," p. 97). In some, memories store descriptions of the objects to be displayed (vectors, letters, numerals), while in others, the memories store a series of points that make up the actual image on the display. At first glance, the image-oriented memory seems most suitable for a raster-scan device, but the object-oriented memory turns out to have more advantages.

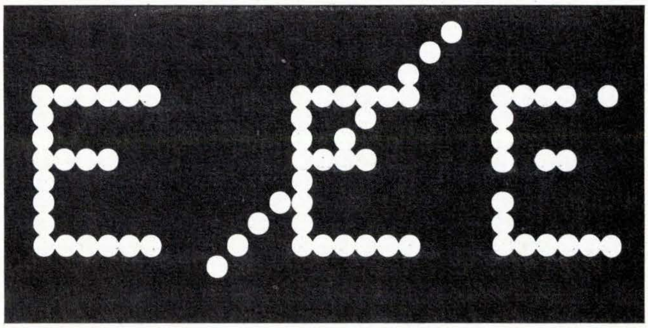
A simple form of object-oriented memory is characteristic of the most primitive cathode-ray-tube displays used with computers. This memory contains a "display list" of words that are successively addressed to control the CRT beam. Changing any part of the image requires merely the changing of corresponding words in the list.

The image-oriented memory, on the other hand, because it contains a point-by-point representation of the visual image, is likely to be much larger than an object-oriented memory. Even so, the use of image-oriented memories is convenient in some display technologies, and it has been encouraged by the radical decrease in cost of semiconductor memories in recent years. Small display memories that cost 10¢ a bit a few years ago are now down to ½¢ to 1¢ a bit and still getting cheaper.

Meanwhile, substantial cost reductions and performance improvements have occurred in other analog and digital circuit elements as well. Moreover, the cost per bit for certain display devices with intrinsic storage is still lower, since the memory is not involved in refreshing an image, only in storing it initially. Some of these devices can hold over a million bits, so that for very complex images—an integrated-circuit mask or a surface-contour diagram, for example—storage displays using image memories are best.

But image-oriented memories lack some of the advantages of the object-oriented display list systems—notably the capability of computer interaction with a displayed object. For example, when two objects cross each other, the object memory "knows" that two points are overwritten at the intersection, and it can remove one without disturbing the other. But when stored in a image memory, the intersection data is lost; an attempt to remove the object can unexpectedly change the other, as illustrated in the sequence shown in Fig. 1.

In dynamic display situations, this problem can be



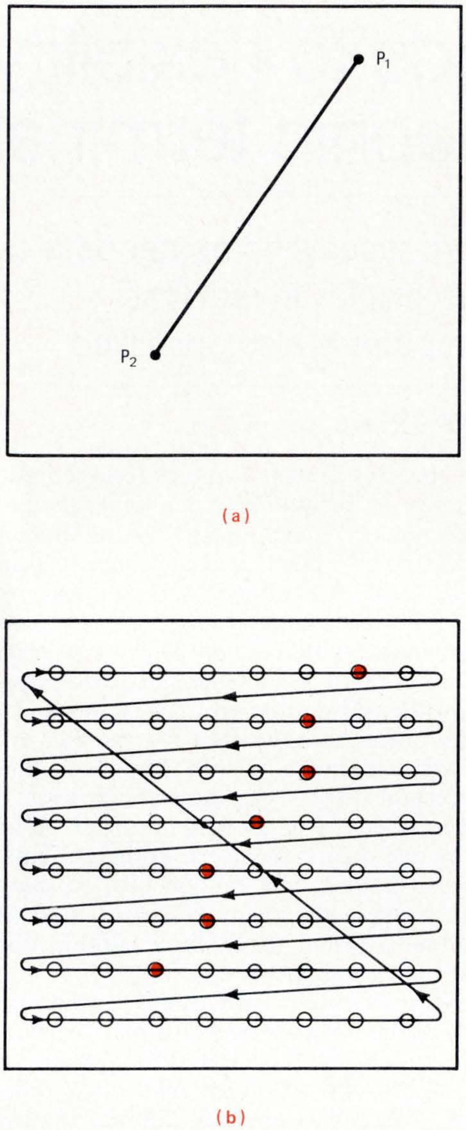
1. Imperfect image. If one of two overlapping images is removed from a display driven by an image-oriented memory, the data at their intersections is lost, and only fragments of the second image remain.

---

*Now with Softech, Inc., Waltham, Mass.

**Now with Tektronix, Inc., Beaverton, Ore.

← Circle 94 on reader service card



(a)

(b)

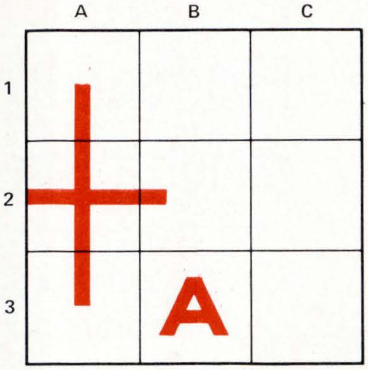
2. Scan conversion. In a raster-scan display, a vector (a) must be converted into a sequence of dots (b). Conversion adds a series of precisely timed intensity pulses to the electron beam so that the dots are properly placed. (On an actual screen, adjacent dots overlap slightly, so that the line appears to be continuous.)

INITIAL DISPLAY LIST

START IN 1A
DRAW VECTOR TO 3A
START IN 2A
DRAW VECTOR TO 2B
START IN 3B
DRAW SYMBOL "A"

REFORMATTED DISPLAY LIST

VECTOR 1A
SKIP TO 2A
VECTOR 2A-1
VECTOR 2A-2
VECTOR 2B
SKIP TO 3A
VECTOR 3A
SYMBOL 3B



3. Memory representation. In a programed-path display, the display list at top left describes the picture. But when the picture is overlaid with the simplified cell structure shown here, the display list is reformatted, as on the bottom left. Items 2A-1 and 2A-2 distinguish between the two different display components in cell 2A.

devastating. For example, to show an airplane moving over a background map, the plane outline is commonly drawn, erased, and redrawn slightly displaced to show movement. For object-oriented representations, this procedure is straightforward, because the lines making up the airplane and the map are stored independently; but for the image-oriented representation, when the airplane outline overlaps the points defining the map, it progressively erases its path across the map. The computer can compensate for this effect, but only at considerable processing expense.

A similar problem is found when data is to be read back from the memory. Characters and vectors in an image memory are only collections of dots, difficult to recognize as single entities, and often impossible to identify uniquely; they provide no indication that they belong together. Therefore, if interaction is needed, there must be a secondary memory to store a display list. To change the picture, the dots associated with the display list items to be removed must be erased, then the dots for the replacement items must be plotted. With an object memory, on the other hand, the display can be altered directly.

For example, text–strings of words–in an object list can be easily edited, while text editing in an image memory requires separate storage and additional manipulation. Similarly, tasks involving blinking or moving vectors, characters or sections of the image, are easy with object memories, but very difficult with image memories. For instance, image-storage graphic terminals usually perform poorly on text editing tasks that alphanumeric terminals, which use object memory lists, handle with ease.

Pros and cons of raster-scan devices

For computer display applications, standard raster-scan monitors have certain obvious advantages. They are by far the most inexpensive general-purpose display components available and come in a variety of sizes, resolutions, and brightness levels. They readily handle gray scale, color, and video mixing. They are the only display devices with a simple, widely accepted interface specification. Monitors of different sizes, wall-sized projectors, hard-copy units, and recorders from different manufacturers can be easily connected together. The fixed beam path and constant beam speed of raster-scan systems require inherently simple compensation and can work with a digital driving source. The display signal can easily be transmitted thousands of feet along a single line to feed multiple display units and can easily be handled by switching systems.

But raster-scan systems also have drawbacks. The

The other types of graphic display

The simplest display systems, used with the earliest computers, generally had two digital-to-analog converters that drove the deflection plates or coils in a cathode-ray tube. A stream of values fed into the converters caused the CRT beam to trace out any desired pattern. These values came from a "display list," which was fed repeatedly to the CRT to maintain or "refresh" the volatile image. This "programed-path" arrangement was simple, reliable, and relatively inexpensive, but also very slow. It could plot only a few thousand points on the screen without annoying flicker—only 50 to 100 characters or a few dozen vectors—and it required the computer's entire processing capability to draw a picture.

From experience with the programed-path display, two things were learned. First, complex pictures that don't unduly burden the processor require special-purpose pattern generators. Second, refreshing the display image should not require processor attention.

By the early 1960s, vector generators, character generators, high-speed CRT beam-deflection systems, and independent display-list memories were being included in display systems. For example, the input to a vector generator might be a word containing the coordinates for one or both end points of a vector, and the generator would translate these into positions on the screen, adding blanking and unblanking signals to turn the beam on and off. Or a word might represent an alphanumeric character with a code of several bits, which a character generator would translate into a series of deflection and blanking instructions for the beam.

With these sophisticated subassemblies, systems could display several thousand characters and over a thousand vectors without flicker. In addition, once the computer had loaded the display-list memory, the picture could be maintained with almost no load on the computer. But these new display terminals cost $100,000 or more, limiting computer graphics to experimental interactions and a few specialized commercial and military applications.

The next big step was to break the cost barrier. Major cost factors were the display-list memory and the high-speed refresh circuitry, both of which were necessary because the displays could only retain images by being constantly rewritten. An image that did not have to be continually rewritten could be generated more slowly, with less expensive circuitry, and would not need refresh memory. These advantages were first embodied in the direct-view storage tube (DVST).

Besides a conventional electron gun, the DVST contains a set of flood guns that keep a written image glowing indefinitely. Consequently, graphic terminals with DVSTs need trace out an image only once. The tube lowered circuit costs, eliminated memory elements, and could display more complex pictures than even the most expensive refresh CRT systems. Prices of graphic terminals fell immediately to less than $15,000, and today such terminals cost less than $4,000—a price drop that has dramatically expanded the number installed.

But the DVST terminal does have some drawbacks. Because of the physics of image storage, for example, brightness and contrast are not as good as in the conventional CRT, nor can the user control these parameters. True gray and color presentations are not possible. Also, DVSTs are complex to build, are available in only a limited number of tube sizes, and are considerably more expensive than regular CRTs.

In addition, while the user can add continuously to the stored image in a DVST, he cannot simply erase parts of an image—instead, he must erase the entire screen and write out a new, slightly different image. Also, the relatively slow erase and writing speeds of the DVST make some operations quite cumbersome, and a dynamic graphic display is next to impossible.

A technology competing with the DVST is the electrical read-out storage tube. Like the DVST, it stores the display image as a charge pattern written by an electron gun. However, it has no flood guns or phosphor. Instead, the stored pattern is sensed but not altered in a "read mode" in which the electron gun scans the image target and feeds the output to a raster-scan monitor.

This tube overcomes most of the disadvantages of the DVST, but it requires two analog subsystems—the deflection circuits and the raster-scan monitor—instead of one. This causes some problems of stability and maintenance, and because the read-out scan only samples the signal, the viewed image is less detailed than the written image.

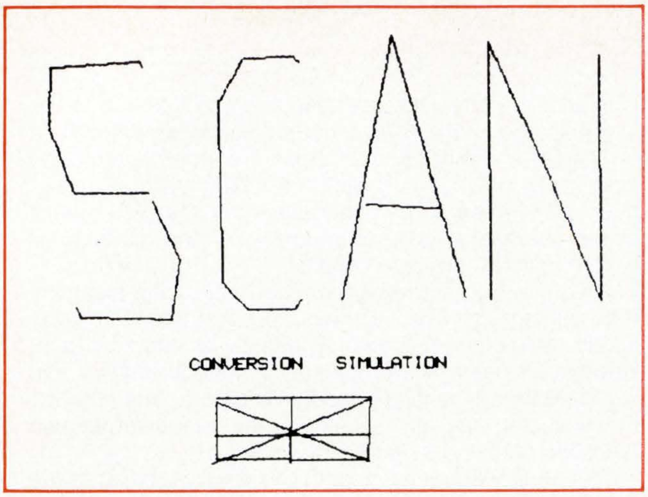
Another commercially successful technique stores the raster signal for the display images on a magnetic disk, from which a monitor is refreshed. The signal is stretched out along one or more linear tracks around the disk.

The most radical departure from previous display technology, however, is the plasma panel, which is a thin sandwich of two flat pieces of glass enclosing a layer of gas. A vertical set of address wires is bonded to one plate, and a horizontal set is bonded to the other. Each intersection of these address line locates a cell. The proper voltages applied with proper timing to one wire in each set ionizes the gas at the intersection, causing it to glow. A different voltage-time combination quenches the discharge. Once the gas is ionized, a sustaining signal keeps it glowing in the absence of the writing signal, so that the panel has a DVST-like memory, but can be selectively erased.

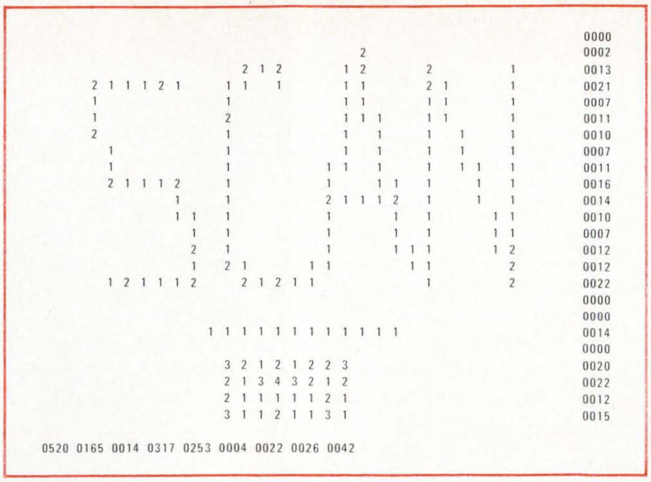
Among other advantages, the plasma panel is digitally driven and provides a crisp distortion-free image. Since it is flat and thin, it can be combined with a rear projector, or the display picture can be copied from the back side of the panel. But the resolution is currently limited to a relatively coarse 70 points per inch. Color and gray scale, though possible, are not at present offered commercially. Worst of all, panel costs are high and unlikely to fall rapidly because extensive circuitry and tight tolerances are necessary.

Meanwhile, the original programed-path refresh CRT systems have not been ignored. On the contrary, they have been given new emphasis for the low-cost market. Programed-path units draw picture elements in an order determined by the computer, so that operations such as enlarging or rotating one element on the screen independently of others are straightforward. These displays present high-quality images and allow maximum flexibility in picture interaction. Light pens can be used to point at objects, which are readily identified by correlating the word being processed with light detection by the gun.

But in such systems the display and its drivers form a complex analog subsystem requiring high-speed signal generation. In addition, compensation circuits are necessary, to correct nonlinearity, pincushion distortion, and problems arising from varying beam speed and fast, random beam positioning. This analog subsystem is still relatively expensive, in spite of declining component prices.



4. Scan simulation. When a reformatted display list, like the one that is shown at the bottom left in Fig. 3, is processed through a simulated display generator, this is the image that results.



```
                                                                                                0000
                                                         2                                      0002
                                    2  1  2              1  2           2              1        0013
         2  1  1  1  2  1        1  1     1              1  1           2  1           1        0021
         1                       1                       1  1           1  1           1        0007
         1                       2                       1  1  1        1  1           1        0011
         2                       1                       1     1        1     1        1        0010
            1                    1                       1     1        1     1        1        0007
            1                    1                    1  1     1        1     1  1     1        0011
            2  1  1  1  2        1                    1        1  1     1        1     1        0016
                        1        1                    2  1  1  1  2     1        1     1        0014
                        1  1     1                    1           1     1           1  1        0010
                           1     1                    1           1     1           1  1        0007
                           2     1                    1           1  1  1           1  2        0012
                           1     2  1           1  1              1  1              2        0012
            1  2  1  1  1  2        2  1  2  1  1                       1              2        0022
                                                                                                0000
                                                                                                0000
                              1  1  1  1  1  1  1  1  1  1  1  1                                0014
                                                                                                0000
                                 3  2  1  2  1  2  2  3                                         0020
                                 2  1  3  4  3  2  1  2                                         0022
                                 2  1  1  1  1  1  2  1                                         0012
                                 3  1  1  2  1  1  3  1                                         0015

0520 0165 0014 0317 0253 0004 0022 0026 0042
```

5. Scan analysis. An analysis program for the simulated display of Fig. 4 lists the contents of each cell, the totals for each scan line, and the total for the whole picture. The picture requires (in octal notation) 520 words of memory, to specify the contents of 253 cells.

biggest has already been mentioned—the fact that converting the normal computer format into the raster-scan signal format is quite complicated. Then, too, the fixed scan-line positions quantize the beam position, so that picture quality suffers, particularly on 525-line monitors. Quality is considerably better on 800- or 1,000-line monitors, but these units are much more costly and require as much as four times the signal bandwidth. Moreover, light-pen operation is more complex, because the timing of the light pen's output is not easily associated with the object being pointed at.

The scan-conversion problem

The beam path in a raster-scan monitor is fixed, and only beam intensity can be varied. To draw a single vector, the system generates a beam-intensity signal consisting of pulses properly spaced in time to show the desired pattern on the screen. More than one vector calls for additional pulses interspersed with one another in a single stream, requiring all vectors to be processed simultaneously.

Consequently, scan conversion begins with a display list of highly encoded words, interpolates the points of each vector between its start and end points, and produces a train of properly timed pulses contributed by all the vector words. The result is a composite pattern on the screen (Fig. 2).

This scan-conversion operation is easier to describe than to implement. To synchronize it directly with the beam's scan position would require each intensifiable screen location to be checked against every word in the display list to see if the beam should be intensified at that location. Since the beam in a 525-line monitor dwells only about 100 nanoseconds at each location, checking every word in the display list of perhaps several thousand words would be extremely difficult.

In the past, the problem has been handled in two steps. First, the display list was processed sequentially, and the pulse trains corresponding to the various vectors were stored in a memory array containing, in its simplest form, one bit associated with each image point. This memory is called a bit-per-element map, or bit map. Second, after the display list had been completely processed, the bit map was scanned, and its output drove a monitor.

However, even though the newest memories have a low cost per bit, bit maps are big—over 300,000 bits for 525 lines in black and white—and therefore expensive. Color pictures, which require three bits for each image point, exceed a million bits per map. A 1,000-line monitor uses even more. By contrast, the original display list may require fewer than 40,000 bits to produce good black-and-white pictures and fewer than 50,000 bits for color or gray-scale pictures.

To regain the advantage of smaller memories and simpler manipulation of object-oriented representations, while retaining the advantages of raster-scan devices, Adage Inc. undertook a three-year feasibility study of real-time conversion from a display list to a raster-scan signal. If an object-oriented display list could be processed 30 times per second, the raster-scan format signal could be fed directly to a monitor without the intermediary of a bit-map memory. But since a whole display list generally cannot be processed this quickly, the approach in the feasibility study was to consider only a small portion of the display list for each location, to keep the processing at a reasonable level.

In any picture produced from a display list, several areas of the screen contain one or more image elements—vectors or alphanumerics. In general, the smaller the area, the fewer the elements (although the relationship is certainly not linear). Therefore, dividing the display list into smaller sublists for portions of the display area reduces the amount of data to be processed for each display location.

The basic subdivision scheme has two parts (Adage has applied for patents on the novel aspects of this scheme). First, the display list is processed into a few dozen sublists which are sorted and stored in the local display-list memory. Each sublist corresponds to a single horizontal band across the picture. If a vector occupies parts of more than one band, it is stored as a number of shorter vectors, each entirely within one band. Since vectors are generally rather short, most of them lie either in one band or across the boundary of two bands. Alphanumerics, on the other hand, are required to be wholly within a band.

Second, the display list for each band is directly converted in real time for display, without using a bit map. Each screen location is checked only against the words in the sublist corresponding to the band containing that location, rather than against the whole display list. In this way, the amount of information processed by the scan-converter is much reduced.

As implemented at Adage, another step is taken between the two described above: the horizontal bands are divided into cells, making a grid of approximately square areas. Preprocessing the display list, more slowly than in real time, generates a new list of short vector segments and symbols, each contained totally within a single cell. These are then placed in the order of the cells on the display, starting from the upper left and ending at the lower right. Now each location is checked only against the sublist for a cell.

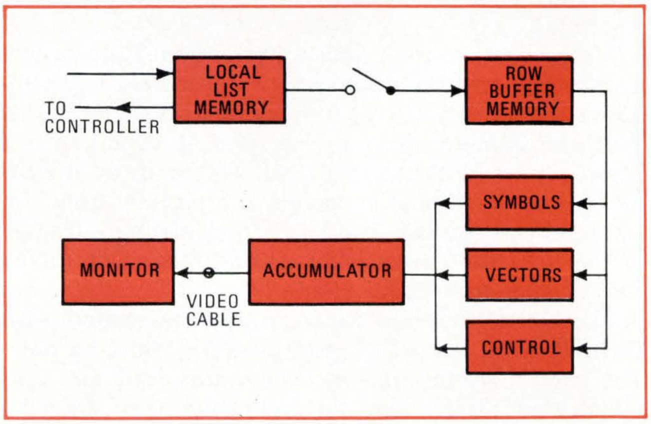
Since the scan conversion is carried out in essentially this order, the segments can be stored in a sequential memory, such as a shift register. The display task then becomes one of scan-converting only the contents of a single cell in real time, rather than processing the entire display list for each intensifiable location.

Making good use of the memory

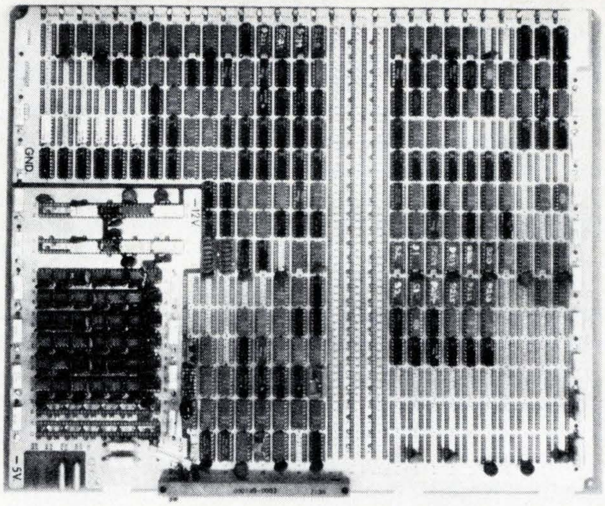
Vectors and characters tend to group, leaving large areas of the screen empty—although these areas are located differently in different images. Thus the memory has to handle both cells with multiple contents, as where lines cross, and cells with no content. Since giving each cell a fixed amount of storage would be inefficient, a quasi-associative memory scheme was chosen.

In this scheme, there are three important aspects. First, some cells may contain more than one item and may need several words to describe them completely. Second, some cells contain alphanumeric symbols, which are stored in the list in the same way that vectors are stored. Finally, address words skip empty cells, which means no memory words are wasted on blank areas of the screen.

The words representing the contents of vector and alphanumeric cells are stored sequentially in memory, using one word of memory for each character or vector segment in a cell. Each word includes a bit that identifies whether the word describes a vector or a character. Another bit can identify the word as an address, blocking both the vector and character interpretation. Only a single address word is required to represent any number of unoccupied contiguous cells. This skip capability averages the storage capacity, so that a simple image covering the whole screen and a highly detailed image in one corner can fit in the same amount of memory.



6. Display generator. The local list memory holds picture information in an object-oriented representation. Vector and symbol generators convert this into raster-scan format, and the accumulator collects the data to feed serially to the monitor.



7. Prototype. Built to amplify results obtained by simulation, this display generator contains a total of 270 integrated circuits, among which are 51 memory circuits and their associated drivers.

For example, Fig. 3 shows a simple display, consisting of nine cells arranged in three rows and containing two vectors and a character. The cells are processed from left to right in each row, and from top to bottom by rows, and the display items are ordered in this sequence in the memory representation.

The first item in the list is the segment of the vertical vector in cell 1A. Since cells 1B and 1C contain nothing, an address word is inserted that specifies no display action in any cell of the top row except 1A. The raster lines, of course, extend across the whole screen, so that the image in 1A in general takes several horizontal sweeps to complete. Then the display skips to cell 2A. In this cell are two different vector segments. Part of the horizontal vector extends into cell 2B. An address word skips cell 2C and, after 2A and 2B are complete, goes directly to cell 3A, which contains the last part of the vertical vector. Finally in cell 3B is a symbol represented by the last word in the list.

Although the display memory's primary purpose is storing the picture, it is a general-purpose read-write memory. Therefore, if the picture being displayed doesn't occupy the whole memory, the rest is available for other uses—for instance, storing additional pictures and storing graphic and text modifiers.

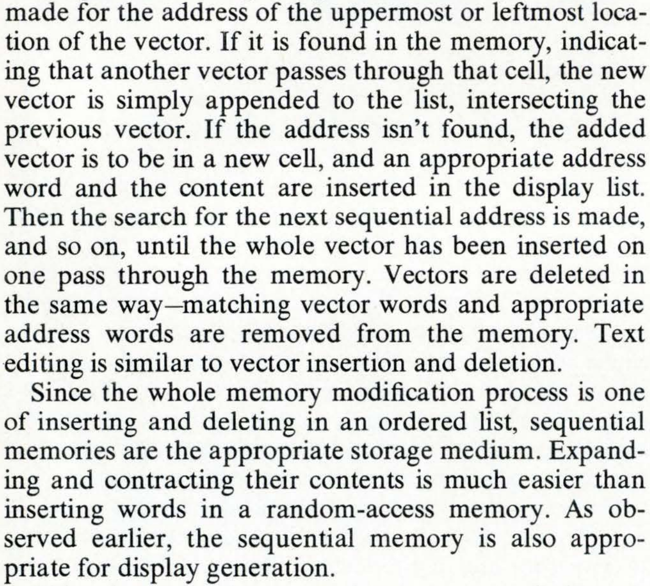
Off-screen pictures can be stored by extending the range of the address words. Each additional bit in the address doubles its range. But for displays of ordinary resolution, the extra range is available without extending the word length just for this purpose, since, in general, vector and character codes require more bits than an address code.

With the extended range, paging and scrolling can be done simply by resetting a "display-top" address instead of reconstructing the whole picture. Information about protected data can be stored within the picture memory itself rather than in an external memory. This information is essentially sequential in nature, just like the cell contents for the picture, so it can be stored in the same way.

Scan conversion is combined with the memory representation in an algorithm for vector and text editing. Suppose, for example, a vector is to be added to an existing image. First, the vector is divided into a sequence of cell contents in order of cell location. Then, starting from the beginning of the memory, a search is made for the address of the uppermost or leftmost location of the vector. If it is found in the memory, indicating that another vector passes through that cell, the new vector is simply appended to the list, intersecting the previous vector. If the address isn't found, the added vector is to be in a new cell, and an appropriate address word and the content are inserted in the display list. Then the search for the next sequential address is made, and so on, until the whole vector has been inserted on one pass through the memory. Vectors are deleted in the same way—matching vector words and appropriate address words are removed from the memory. Text editing is similar to vector insertion and deletion.

Since the whole memory modification process is one of inserting and deleting in an ordered list, sequential memories are the appropriate storage medium. Expanding and contracting their contents is much easier than inserting words in a random-access memory. As observed earlier, the sequential memory is also appropriate for display generation.

This scheme has been tested on some actual pictures, covering many applications, on a simulator that was programed with a direct-view storage-tube terminal connected to a minicomputer.



8. Complex display system. A display can consist of as little as a computer, a generator, and a monitor, or it can be much more complicated and take varied forms like those shown here. DG and DS stand for display generator and display station, respectively.

Each picture was stored as a list of cell contents in the minicomputer memory. This provided a check of the software necessary to preprocess a picture. Programs were then written that simulated the functions of the vector generator and symbol generator of the real-time scan-converter hardware. These programs processed the cell contents and produced the dot-by-dot scan representation of the picture, which was then plotted on the display (Fig. 4). Finally, an analysis program plotted the number of elements in each cell, and totals for each row and for the whole picture. This verified theories on picture density, and permitted an estimate of how much memory the picture would require (Fig. 5).

The simulation had several useful results. It demonstrated, long before hardware was built, that the scheme worked, and gave an impetus to further development. It showed that, as had been anticipated, the cell boundaries were not apparent in the pictures. It proved that significant pictures could be stored in 2,000 words of memory. It suggested several variants on the hardware and software schemes, which were tried out. And it permitted experiments with different cell sizes, to find the most efficient size, various memory representation schemes, and different combinations of hardware and software vector and symbol generator techniques—all before any hardware was built.

The experimental display generator

Once the display generator was completely specified as a result of the simulation, a hardware prototype (Fig. 6) was necessary to drive a raster-scan monitor. No computer could simulate the scan-conversion algorithm fast enough to generate a real-time raster-scan signal.

In the prototype, a local list memory stores item and address words that describe individual cells. All the words associated with a particular horizontal row of cells are loaded into the row buffer memory, and thence fed to a control element and to either a vector or a symbol generator. While the memory is feeding the data out, the list memory is disconnected, and a controller can read, load, or reload it without interfering with the scan-conversion operation and without causing the screen to flicker.

The symbol generator uses a conventional approach—a read-only memory containing a dot matrix representation of the various characters. However, the vector generator is rather more complicated, to cope with the thousands of different vectors that can be drawn in cells of reasonable size. It generates the pattern from the start and end points in the cell, by calculating the slope of the line, and successively adding this value to the start position on each scan line.

The control section translates bits associated with each list word into color, gray scale, or blinking data. The control section and the generators feed an accumulator which combines the contributions of the various words until the section of the scan line within the cell is completely described. When it is complete, the pattern is dumped serially to the raster-scan monitor while the next cell is processed. An averaging circuit between the accumulator and the monitor (not shown on the block diagram) smoothes out the variable processing rate of the generators and supplies a continuous drive signal for the monitor.

The prototype board is shown in Fig. 7. It runs on a 6.3-megahertz clock and contains 270 integrated circuits, including 51 MOS memory circuits and drivers. Besides the memory and raster-scan conversion logic, the board also contains keyboard circuits, and a cursor generator. Most of the logic was designed with standard TTL components; a few Schottky elements were used. The system contains no adjusting potentiometers, except those in the monitor.

System considerations

To be useful, a display generator must be part of a system. The most straightforward system is a display generator driving a monitor under the control of a computer. In more complex applications a powerful (and expensive) computer supports multiple terminals in any of several configurations. Some of the possibilities for a system are shown in Fig. 8.

Many large systems avoid cluttering up the primary processor and improve response time by incorporating satellite processors to handle trivial tasks such as data editing, input-output transfers, and communications control. The Adage experimental system includes a satellite processor, which can readily divide vectors into cell contents. It can also process keyboard inputs, move the cursor, and do local editing.

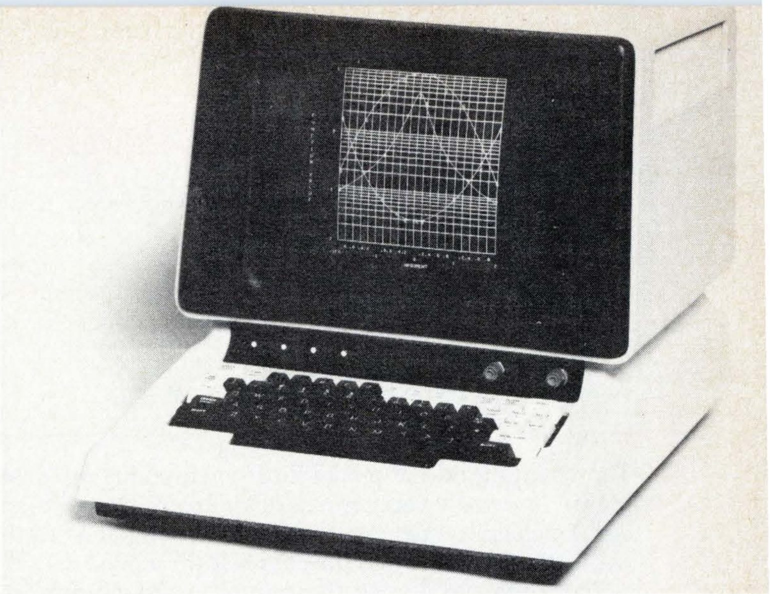
The experimental station (Fig. 9) consists of the monitor and a keyboard, which contains cursor keys, editing keys, transmission keys, and general function keys, in addition to the standard alphanumeric keyboard. Cursor keys move an alphanumeric cursor (underline) or a crosshair graphic cursor. Editing keys signal both character and line-editing functions. Transmission keys cause edited text to be sent to the host computer. For flexibility, the general function keys may be interpreted and processed by the local computer, or sent on to the host computer.

Doing better

The system can be readily expanded. Features such as color or gray scale are obtained by adding a few extra bits to the display memory words, and are implemented by expanding the assembler and video signal generator.

An impressive feature, derived from the raster-scanning principle on which the system is based, is the capacity to show shaded background regions. This is achieved by specifying, for an individual vector, the intensity or color not only for the individual points through which it passes, but also for all points to its right on each scan line that intersects it. Thus a shaded area can be specified by adding an extra bit to the vectors along its left side to specify shade, and then letting the other vectors along its right side set the normal background. The assembler gives the shading intensity or color to each element on the scan line that is not otherwise specified by foreground information.

The use of standard video signals opens the way for combining a raster-scan terminal with a variety of television-based components such as TV-projection equipment, video-tape recorders, cameras, and hard-copy devices. These pieces of equipment can be interconnected with almost no special engineering, because the television signal standard is complete and widely accepted.



9. Display station. Twelve-inch raster-scan monitor is able to present both graphic and alphanumeric data. The keyboard provides cursor control as well as text-editing functions.

Of particular interest is equipment that mixes video signals from different sources. For example, computer-generated images and microfilm or slide images can be picked up by a video camera and displayed simultaneously on the same terminal. Or signals from two or more display generators can be mixed. In this way, one generator can be made to display background information, while others produce changing data. Alternatively, data from multiple display generators can be combined in images with a complexity beyond the capability of a single generator.

A further advantage of the raster-scan scheme is its adaptability to higher-resolution monitors by somewhat more complex circuitry. Among other things, such monitors require larger cell sizes and more parallel operation because of the increased beam speed.

What lies ahead

The very powerful and very expensive graphic systems will continue to have an important place in high-performance command-and-control systems. Image-oriented systems will be used to produce complex, high-density pictures requiring little interaction. But for much interesting and practical work, real-time scan conversion of object-oriented displays offers an attractive alternative for graphic-system users.

Graphic displays have long been more promising than practical, because the equipment has been expensive, software support has been inadequate, and potential users have simply not been aware of the possibilities. Now technology is turning the tide with low-cost and high-performance equipment, a growing storehouse of applications programs, and increasing sophistication among computer system users. During the next few years graphic terminals will probably appear in more and more purely commercial roles, rather than in just experimental environments. ☐