Toward a Machine with Interactive Skills

While Boston and its environs slept away the predawn hours of a November morning in 1961, Ivan Sutherland, doctoral candidate in electrical engineering at the Massachusetts Institute of Technology and computer programmer extraordinary, sat engrossed at the console of the TX-2 computer in the basement of M.I.T.'s Lincoln Laboratory. He was at work on his dissertation project, a computer drawing program he called Sketchpad. The name derived from the proclivity of engineers to rough out an idea on a scrap of paper, then gradually refine it by making innumerable revisions. Sutherland was convinced that he could turn the computer into a superior tool for this process.

Sutherland was only one of many researchers captivated by the interactive possibilities of the TX-2, so he had to scramble for a share of the computer's precious time, even if that meant getting out of bed at two or three in the morning. The solitude of his sessions at the computer did not bother the 23-year-old Sutherland; in some ways it suited his personality. Brilliant and idiosyncratic, he was known as a man who went his own way. He could focus his formidable powers of concentration and shut out anything extraneous. "It wasn't that he was unfriendly," an acquaintance explained; "he just zeroed in on whatever he was involved in. If you were at his house and it was time for him to go to bed, he would get up and excuse himself and tell you to lock up when you left. That was just his way."

On that November morning, Sutherland's left hand hovered over a box studded with closely spaced push buttons. In his right he held a light pen that resembled the light gun invented more than a decade earlier and used on the Whirlwind computer. Sutherland held the tip of the light pen to the center of a computer monitor, where he had programmed the word "ink" to appear. At the touch of the pen, the word was replaced by a small cross. Sutherland then

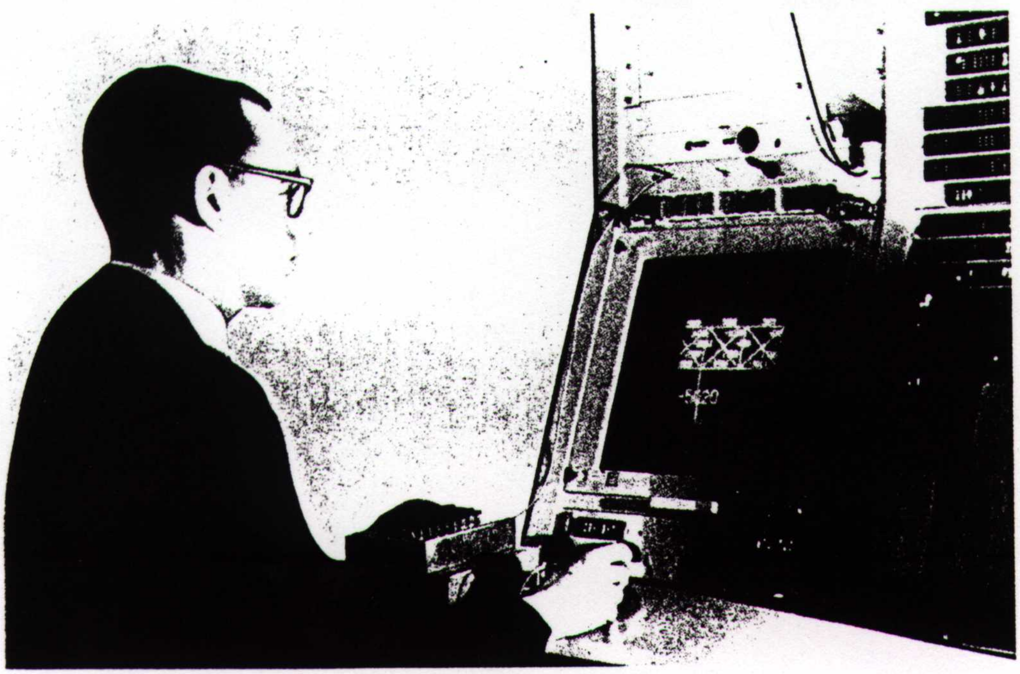
Stripped to structural basics, an Air Force F-15 jet fighter is shown as a three-dimensional wire-frame model on a high-resolution vector monitor. Vector displays are used by designers and engineers in many fields to isolate and clearly identify components.

punched one of the push buttons and began to move the light pen. As he did so, a bright green line appeared, stretching from the center of the cross to the pen's new position. Wherever he moved the pen, the line followed, like a rubber band with one end tacked to the center of the cross and the other attached to the pen. With a second push-button command, the line remained frozen on the screen as Sutherland moved the light pen away.

Modest as it may seem a quarter of a century later, the line was proof of an extraordinary accomplishment. In a stroke, Sutherland had extended the practical applications of interaction between human and computer. Sketchpad was a masterly synthesis of two different branches of developmental effort. The first was work done by programmers on the TX-2 and its predecessor, the TX-0, to develop software that gave these machines limited line-drawing capabilities. Sutherland also utilized programming techniques invented for the APT (Automatically Programmed Tools) System, an early computer-aided manufacturing system that used a computer to control tools for milling pieces of equipment.

Realizing that computer graphics could have significant applications in engineering and design, Sutherland bent his efforts toward increasing the operator's control of the image on the screen. Where an operator at the TX-2 could draw simple shapes on the screen with a light pen, the shape could not be manipulated. "In the past," Sutherland explained in the technical report on Sketchpad that M.I.T. published in January 1963, "we have been writing letters to rather than conferring with our computers."

In contrast, Sketchpad promised to turn the computer into a tool that anyone might use. Even at this early stage, the program allowed someone with no programming experience to solve complex engineering problems through the



Sitting at the console of the Lincoln Lab TX-2 computer, programmer Ivan Sutherland uses the Sketchpad graphics program to manipulate a bridge design. This pioneering interactive program, devised by Sutherland in 1961, enabled users not only to draw and erase on the screen but also to demonstrate the results of engineering tests. Commands were entered with the light pen in Sutherland's hand and the push buttons to his left.

use of computer graphics. In effect, Sketchpad translated the operator's wishes into the computer's binary language and displayed the machine's response instantly, in real time.

Sutherland's work at M.I.T. would ultimately be viewed as heralding a crucial turning point in the history of computer graphics. Before Sutherland and Sketchpad, most of the graphics applications of computers were for military purposes and were relatively crude. After Sutherland, computer graphics was used increasingly as a tool for industrial engineering and design, primarily in the auto and aerospace industries but gradually in others as well. Largely because of Sutherland's skill and vision, computer graphics moved into the world beyond the laboratory and the military base.

EN ROUTE TO A SIMPLE LINE

Sutherland had first decided to write a computer graphics program in April of 1961. The preceding winter, while working on other projects, he had become familiar with the TX-2 and noticed that its design made drawing convenient. Besides the light pen, it had a CRT screen and enough memory to hold 280,000 bytes, a prodigious capacity at the time. Furthermore, the TX-2 could be modified with little difficulty, and Sutherland requested the addition of a bank of push buttons. Work on Sketchpad began in earnest in the fall of 1961. An early success was to make a small cross displayed on the computer monitor follow the light pen obediently around the screen. The cross provided a signal for the light pen to detect and lock onto, and served as the starting point for a drawing. To accomplish this, Sutherland wrote a program instructing the computer to draw the cross, centered on the pair of coordinates *(pages 10-11)* closest to the tip of the light pen. To make the cross respond to the pen's movement, the program analyzed signals generated when the pen touched any part of the cross; the signals determined where the pen lay in relation to the cross's center. If the pen had moved, the program redrew the cross, centering it at new coordinates nearest to the tip of the pen. Repeated hundreds of times each second, these steps took place so rapidly that the cross appeared to tag along after the light pen as the pen was moved from one point to another on the screen.

Sutherland had to program a score of other details before Sketchpad could draw a simple straight line. For example, he wrote instructions to the computer that interpreted pushing a button as an order to remember the coordinates of the cross's position at that instant. Another subprogram directed the computer to calculate a new set of points lying on a straight line between the initial fixed point and the point to which the light pen might be moved. This subprogram also interpreted a second push-button command as an instruction to end the line. A third routine stored the line in a portion of the TX-2's memory called the refresh buffer. The computer retrieved lines from the refresh buffer and redrew them 20 times a second to keep them from fading.

Sutherland was just getting started. Even though Sketchpad could draw straight lines, it knew nothing of curves, essential forms if the program were ever to be of service to engineers and draftsmen. Sutherland first simplified the problem by limiting the program's curves to circular arcs. To draw an arc or a complete circle, the operator would tell Sketchpad with one push-button command where the center of the circle was to lie, then move the pen and push the button a second time, thus defining the length of the radius and telling Sketchpad where to begin drawing. At that point, the program would produce an arc until a third command halted the process.

Not until mid-1962 could Sketchpad draw a portion of a circle that was larger than the screen—a feature necessary for making shallow curves. In effect, Sutherland had to make the edge of the screen disappear for the computer. He did it by establishing phantom coordinates beyond the screen's perimeter. As long as the radius of a circle was less than the width of the screen, the TX-2 could draw the circle, although only the part that fit on the screen would be displayed.

DEFINING OBJECTS WITH RULES

Still to be programmed into Sketchpad were routines that Sutherland called constraints, the rules for manipulating a drawing. At this stage in the program's development, drawing a shape such as a square was akin to laying out matchsticks on a table. Even if the operator could draw four lines of equal length, arranging them perpendicular to one another was difficult. Nonetheless, the result was merely four lines in the shape of a square; repositioning the square would ruin it as surely as if it had indeed been made of matchsticks.

Sketchpad's constraints allowed an operator to create an object by defining its properties. A square might begin as a rough, four-sided figure, but the operator could then instruct the program always to keep the lines joined in a closed shape, to make them equal in length and to keep all four angles locked at 90 degrees. Thereafter, if one side of the square was moved on the screen, the other sides would be forced to come along in order to obey the constraints associated with that shape.

These rules and others that Sutherland later programmed into Sketchpad were to give computer drawing its great potential. A Sketchpad drawing would consist of points, lines and arcs linked together to form objects. Once created, any object, or shape, could be enlarged, reduced or rotated on the screen. It could be stored in a library of shapes and recalled for future use. It could be duplicated and the copies arranged to generate a new shape. With these rules, a complex drawing of a bridge or an automobile could be created as a hierarchy of simpler components, each of which could be altered and moved around on the screen.

As is often the case with innovative software, little of Sutherland's programming operated perfectly at first. "The trials of getting systems to work are many," he wrote later. One memorable trial concerned rotating an object on the screen. If, for example, a small figure were drawn inside a larger one and linked with it to make a single object, then rotating the outer image to the right should have caused the inner one to turn in the same direction. Perversely, the small figure revolved to the left. After tedious hours of debugging, Sutherland unmasked the problem: He had transposed some positive and negative values.

"By Memorial Day, 1962," continued Sutherland, "it was possible to see that sketching could indeed be done on the computer." One early test was to produce graph paper based not on squares but on hexagons. Within an hour after beginning work, Sutherland had linked 900 identical hexagons edge to edge, and a flat-bed plotter connected to the computer had drawn a 30-inch-by-30-inch master of the pattern. "Professional draftsmen," noted Sutherland, "estimated that it would take them two days to produce a similar pattern." Sketchpad was at last demonstrating the speed and agility that Sutherland had envisioned.

AN INSPIRATIONAL EFFECT

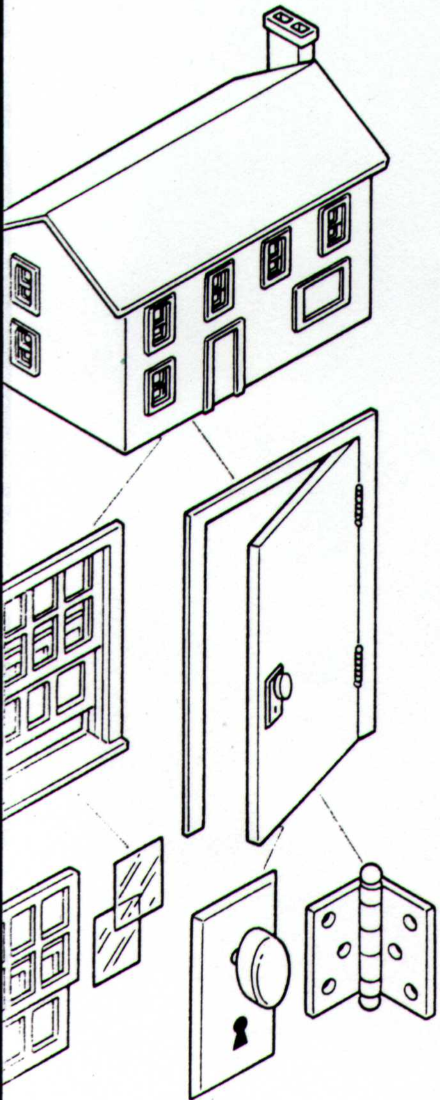
About this time, Sutherland made a movie of Sketchpad in operation. He showed it around the M.I.T. campus to students and teachers, soliciting their opinions. Copies of the film made their way to other campuses. David Evans, who later joined Sutherland in founding a computer graphics company, was then a computer science and electrical engineering professor at the University of California at Berkeley. He remembers that the Sketchpad film was "like an underground movie. There were lots of copies around. It was immediately obvious that this was beyond what anyone else had done. It was elegant."

The Sketchpad film debuted at the spring Joint Computer Conference in Detroit in 1963, where it fired the imaginations of those who had not yet seen it. Andries van Dam, later chairman of the Brown University computer science department, was one of them. He decided "on the spot that graphics was going to be my research area." Van Dam shared Sutherland's opinion that Sketchpad's most immediate value lay in designing new products. An electronics engineer, for example, might use Sketchpad to create a library of basic circuit components such as transistors and resistors, then combine them into amplifiers, motor-control circuits and other useful devices.

Nevertheless, the reaction from potential corporate sponsors was disappointing at first. Industry in general seemed to have something of a blind spot when it came to computer graphics, regarding the idea as a bit of impractical electronic razzmatazz. One man who had observed it all from the start—and had experienced the frustrations—was Thurber Moffett, an aeronautical engineer for General Dynamics Corporation in San Diego. In 1955, Moffett had collaborated with an IBM engineer named Robert Cortney to develop an engineering computer system for the Defense Department. Moffett and Cortney spent five months in San Diego working up a proposal. One of their aims was to show engineering data as graphs on the computer screen. But evenings after work, they speculated about matters that went beyond the task at hand. "Bob and I used to have dinner together two or three nights a week at the King's Inn," Moffett recounted, "and we diagramed what we thought an interactive computer graphics system should look like on the tablecloths. We did it just to amuse ourselves. We had no real notion what would come out of it."

Moffett remembered the ho-hum reaction from corporate executives who saw presentations of Sketchpad: "They would look at it and go away thinking that it was some witchcraft, something they did not have to worry about until maybe 2010. There was probably not one in a thousand who realized what Ivan had done."

Among the few who appreciated Sketchpad were the executives of General Motors Corporation. GM, in fact, had been pioneering the idea all along. In 1959, two years before Sutherland began work on Sketchpad, GM entered into a farsighted collaboration with IBM to build a computer system that would facilitate the design of cars. The project was called DAC-1, for Design Aug-



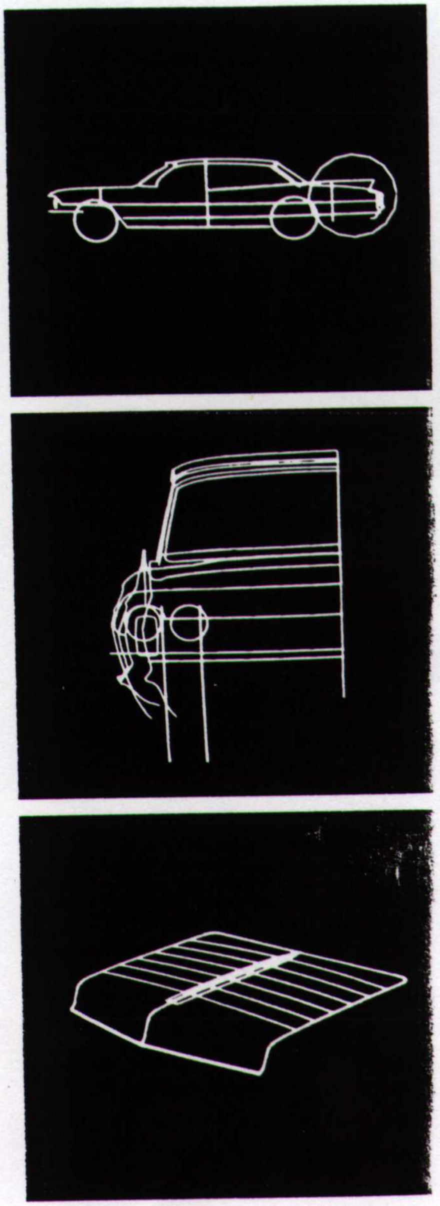
A graphic data base—the computer files containing all of an image's basic elements—is frequently organized in the so-called tree structure shown here. Conceived by Ivan Sutherland for his Sketchpad program, such data bases allow quick access to information by storing data in major sets, each containing subsets. For example, to call a door hinge onto the screen the operator would look for "door" as a major set within the "house" data base, and "hinge" as a subset within "door."

mented by Computers. Like Sutherland, DAC-1's creators wanted to turn the computer into a design tool, but as is common in the world of computers, they arrived at a different solution.

DAC-1 was unveiled in 1964 at the Joint Computer Conference, the same forum that Sutherland had used for Sketchpad a year earlier. Where Sketchpad's repertoire of curves was limited to arcs of circles, DAC-1 could reproduce the flowing lines favored by automobile stylists, even though such curves obey no simple mathematical formulas. But because a car's shape must be drawn more precisely than a draftsman can work freehand, DAC-1 contained no provision like Sketchpad's for creating a design on the screen from scratch. Instead, the system required that a designer either describe the desired shape in a program and feed it into the computer or submit conventional engineering drawings to be read into memory with a digitizing camera. With the drawing in the computer, DAC-1 became more interactive, employing an electronic stylus, much as Sketchpad used a light pen, to permit the operator to enlarge the drawing and change portions of it. GM shied away from describing the system as being able to perform in real time, but in fact, DAC-1 reacted quickly.

Thurber Moffett recalled that people lined up two hours ahead of time to get in to see the DAC-1 demonstration. The technicians operating the machine "showed a structure rotating on the screen," Moffett said. "Nobody had ever seen that before." As Moffett walked from the conference rooms after the demonstration, he heard someone call his name. It was Robert Cortney. Moffett asked his old friend if he had been involved in the GM graphics system. "That's my baby," Cortney replied.

"You son of a gun," Moffett exclaimed, "you saved those tablecloths!"

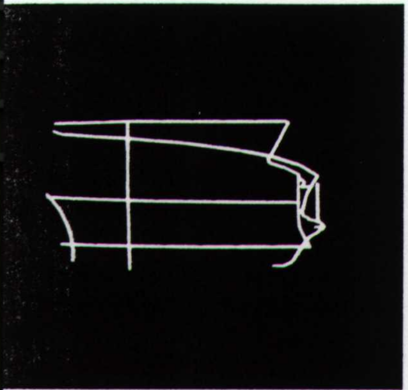


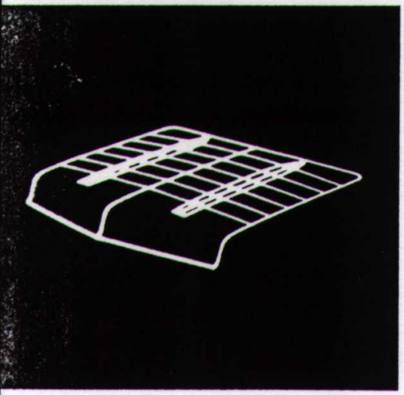
Frames from a 1964 demonstration film show the capabilities of DAC-1, the revolutionary computer-aided design system developed by General Motors and IBM. With the touch of an electronic pencil, the designer of this Cadillac could enlarge a circled area (*top two frames*), rotate the model (*middle*) or modify the design of the trunk (*bottom*).

GM'S SEAL OF APPROVAL

General Motors' commitment to computer graphics proved to be a breakthrough; if a company as big and smart as GM was putting its money on the line, computer graphics had to be more than just an electronic flash in the pan. Lockheed-Georgia went to work on a graphics system to help in the design of airplanes. General Electric, Sperry Rand and TRW all showed interest in exploiting the newly discovered graphic powers of the computer. Oil companies also began work on computer systems for charting data derived from exploratory seismic soundings. By the end of 1964, estimated Carl Machover, a computer graphics consultant, there were perhaps 100 computer graphics terminals in use at various companies and universities. But like DAC-1, they were all one-of-a-kind, custom-designed systems. A company that wanted a ready-made terminal was out of luck.

That situation changed in 1965 when IBM, noting the surge of interest in computer graphics for design and engineering, capitalized on its experience with DAC-1 to bring out the first commercially available computer graphics terminal. Called the IBM 2250, it was built to work with the company's new 360 series of mainframe computers. Whereas DAC-1 was tailored for the design of automobiles, the 2250 was an easily adaptable general-purpose machine. IBM's commitment gave the nascent field of computer graphics much-needed credibility. One immediate effect was a surge of competition for the business of providing graphics terminals to use with IBM's and other manufacturers' mainframe computers.





A front runner was Evans & Sutherland Computer Corporation, a Salt Lake City company founded in 1968 by David Evans and Ivan Sutherland. After earning his doctorate from M.I.T., and before joining the faculty of Harvard in 1966, Sutherland had served two years as chief of the information technology program at the Defense Department's Advanced Research Projects Agency (ARPA) at the Pentagon. Evans was doing work for ARPA at the time, and the two formed a vision of what computer graphics could become. Ultimately, Evans & Sutherland would design and sell scientific and engineering workstations and flight simulators for training airline pilots. But one of the first projects the new company tackled was to build a computer graphics terminal that could outperform the IBM 2250.

A QUESTION OF SPEED

Evans & Sutherland took aim at the 2250's major limitation. IBM's system was designed to operate as a terminal relying solely on a graphics program loaded into a mainframe computer. That arrangement, while suitable for tasks that did not require the undivided attention of the computer's central processing unit, limited graphics performance in two ways. First, the software operated too slowly to handle complex drawings such as those for mechanical engineering. In any graphics system, speed is an issue because the computer has to redraw the image continually; if there are more lines on the screen than the computer can redraw in the $\frac{1}{30}$ second or so allotted to the task, lines dim and the image flickers. Second, the rotation process took up too much computer time. To rotate a drawing on a terminal screen, the graphics software in the host computer calculated a new position for each line before redrawing it. If the calculations were interrupted, the rotation would become jerky; thus, turning an object around on the screen monopolized the computer, taking precedence over tasks the machine might be asked to perform by other users.

To avoid these two pitfalls, Evans & Sutherland planned to do something radical. The company decided to build special-purpose digital circuits into the terminal of their LDS-1 (for Line Drawing System). These circuits would rotate objects and refresh the screen, thus relieving the host computer and software of those tasks. Evans & Sutherland built a special processor that significantly shortened the time required to refresh the screen, increasing by as much as a factor of 100 the number of lines that could be displayed before flicker set in. This also enabled the machine to change the image on the screen with amazing speed. The LDS-1 astounded Stanley Ruggio, a General Dynamics Corporation engineer who saw it in 1972. "Its speed of putting up the display," he reported, "was awesome—half a second, almost instantaneously."

For all its wizardry, the LDS-1 did not address one issue: the towering cost of computer graphics. It was a bargain, certainly, considering what it could do, but it cost $250,000, twice as much as the IBM 2250. For either machine, the necessary graphics software cost a like amount. In the face of such prices, even large companies that already owned or leased mainframe computers often balked at the cost of adding graphics to the system.

One way to lower the price of a graphics terminal was to curb its voracious appetite for memory in the refresh buffer. For every line in a drawing, the computer had to remember a set of coordinates for each of the two endpoints; for an arc of a circle, three points; and for a free-form curve, hundreds of points. With each point in a three-dimensional image needing eight bytes of storage, the memory requirements for a detailed drawing of an aircraft, say, were astronomical. If these requirements could be reduced or eliminated, a graphics terminal might be built for a fraction of the cost of the Evans & Sutherland device.

As it happened, a device that could break the memory price barrier had been around since 1965, introduced by engineers at an Oregon firm called Tektronix, a builder of oscilloscopes and other electronic test instruments. The direct-view storage tube, or DVST, was much like an ordinary vector graphics CRT, except that it contained a mesh of fine wires behind the phosphor coating on the screen. As the electron beam darted about the screen to draw an image, it charged the wire grid with electricity that kept the image aglow for as long as an hour, making high-priced refresh buffers unnecessary. The cost difference was dramatic. Where a graphics system with a refresh buffer often operated for as much as $250 per hour, a terminal with a direct-view storage tube could operate for $10 to $30 an hour. "The storage tube made graphics affordable for the masses," said Andries van Dam of Brown. "You could hang them as terminals over telephone lines to mainframes and all of a sudden you had el cheapo, poor man's graphics."

There were, to be sure, liabilities, which was why Evans & Sutherland had opted not to use the storage tube in the LDS-1. For one thing, the DVST could produce only two-dimensional images. Furthermore, the image was slow to come up on the screen, and when it did, it was fuzzy; lines were so faint that room lights had to be dimmed for the lines to be visible. More important, selective erasing and rotation were impossible.

Nevertheless, the DVST filled a definite need. Not everyone who could benefit from computer graphics required—or could afford—the sophisticated features of an LDS-1. For example, scientists analyzing graphed data or plotting topographical maps did not need real-time response or the ability to rotate images. Ultimately, Tektronix set the price for DVST terminals at a mere $4,000 each, and thousands of them were sold over the next decade. Until a radical decline in the price of memory made practical a new technological twist, DVSTs would dominate the graphics field.