SKETCHPAD III, THREE DIMENSIONAL GRAPHICAL COMMUNICATION WITH A DIGITAL COMPUTER

by

TIMOTHY EDWARD JOHNSON

B.S., Univ. of Michigan
(1961)

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

at the

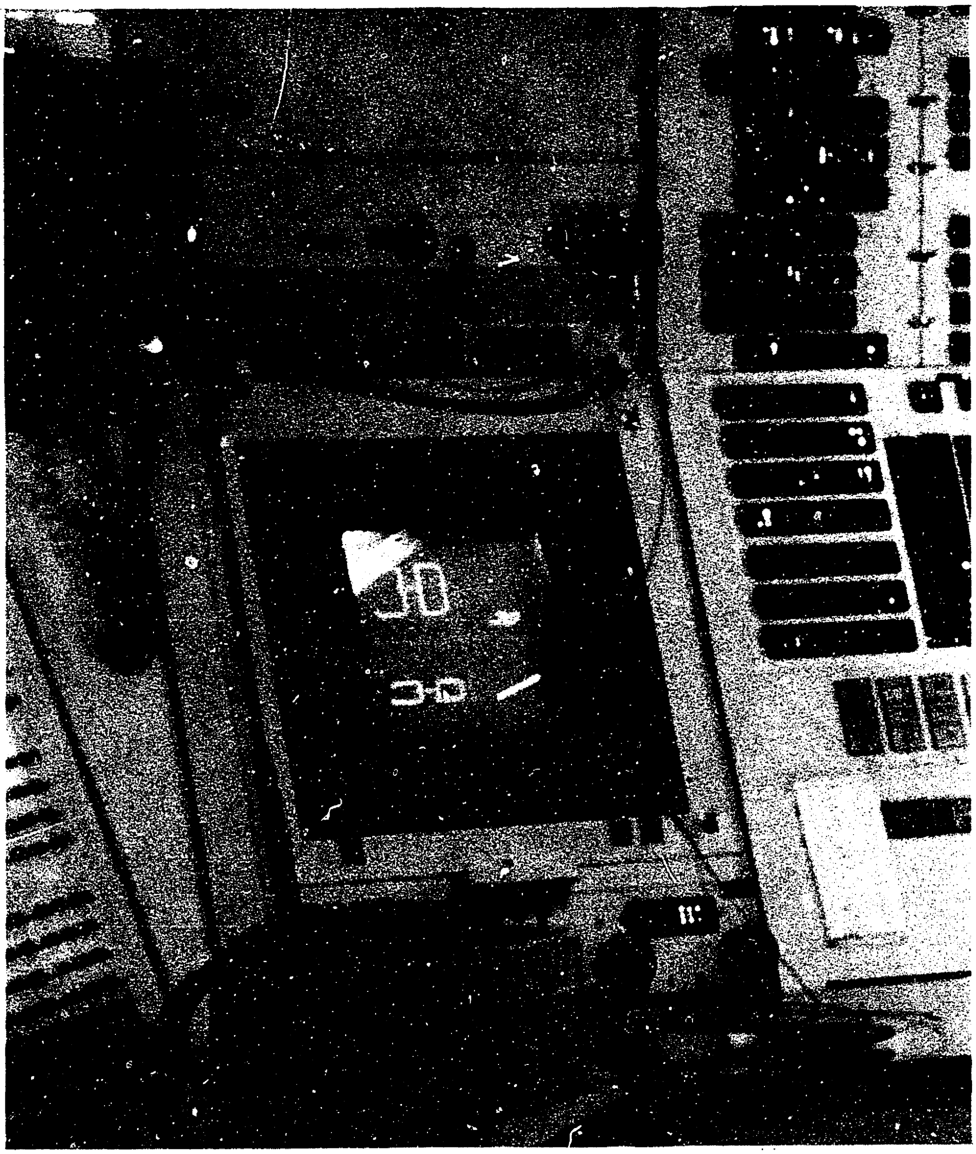
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
May, 1963

Signature of Author Timothy E. Johnson
Department of Mechanical Engineering, May 17, 1963

Certified by Steven Anson Coons
Thesis Supervisor

Accepted by WM Rohsenow
Chairman, Departmental Committee on Graduate Students

√



TX-2 Console Area

Directly below the CRT display of the letters "3-D" (ceiling-light reflections show) are four digital shaft encoders used to modify the display. The light-pen lying on the table to the right is used for drawing. The pushbutton console is the box on the left side of the table. Rows of toggle switches used to select different displays appear along the left edge of the photo.

038

SKETCHPAD III, THREE DIMENSIONAL GRAPHICAL COMMUNICATION WITH A DIGITAL COMPUTER

by

TIMOTHY EDWARD JOHNSON

Submitted to the Department of Mechanical Engineering on May 17, 1963 in partial fulfillment of the requirements for the degree of Master of Science.

ABSTRACT

For effective graphical communication in Computer-Aided Mechanical Design, the ability to construct and manipulate three-dimensional figures is required. This paper describes a programming system which permits scope, light-pen and control knobs to be used in a flexible manner to draw three-dimensional, straight line, "wire frame" figures. Three orthogonal views complement a perspective view to permit simultaneous observation from several vantage points to increase depth perception.

Thesis Supervisor: Steven A. Coons

Title: Associate Professor of Mechanical Engineering

ACKNOWLEDGEMENTS

I am indebted to Professor Steven A. Coons of the Mechanical Engineering Department for his help and advice. In his modest fashion he gave guidance at critical periods and offered many suggestions which have been embodied in Sketchpad III.

The interest and discussion offered by the people of the Computer Aided Design Project has been greatly appreciated.

Special thanks go to the M.I.T. Lincoln Laboratory for its support and interest. I thank Wesley Clark, Jack Mitchell, Jack Raffel, and Dr. Frederick Frick for making the TX-2 computer available to me. I am grateful to Ivan E. Sutherland and Lawrence G. Roberts for their patient instructions and suggestions on how best to use the TX-2. I am further indebted to the two of them for the computer subroutines and programs they unselfishly gave me which greatly accelerated this research.

Thanks to the Engineering Projects Laboratory of the Mechanical Engineering Department which provided me with office space and an enlightening environment.

Final thanks go to my wife, Ann, who patiently waded through my manuscripts correcting this engineer's poor grammar. The design shown in Fig. 12 is based on an original chair design created and built by my wife.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF FIGURES

CHAPTER I

SYNOPSIS

A. INTRODUCTION

The standard computer input-output system of today, which allows man to interact with the machine, forces man to reduce his communications to written statements suitable for typing. This method is too slow for conversing with the computer; it stifles real-time, bilateral communication. Furthermore, written forms are too cumbersome for many kinds of information.

The Computer-Aided Design Project at M.I.T., faced with the problem of rapidly communicating structural objects to the computer, needed something other than written languages to describe machine parts and space frames. The Sketchpad III system, developed out of this need, allows man to communicate graphically simple three-dimensional objects to the digital computer. The system does not read blueprints; it is fashioned to be an active partner in the design process.

In design work, man and computer must rapidly communicate with each other. The design of a mechanical or structural object begins with a graphical description. Initially this description is not a precise statement of refined detail, but is a vague stirring of the imagination. This concept is developed into the finished design through modification, deletion, and analysis.

Thus, the computer system, as a partner to the designer, accepts, interprets, modifies, and remembers shape description information introduced graphically. Graphical communication between man and machine is not similar to pencil and paper methods. The digital computer permits the adoption of new graphical techniques, but these new methods are not so far removed from "drawing board simplicity" so as to render the system inconvenient. For instance, Sketchpad III allows the designer to rotate a shape description to any attitude in space. This permits an object to be placed in the best orientation for graphical manipulation or visualization.

General three-dimensional graphical communication, which deals with arbitrary surfaces and space curve intersections, presents many difficult problems; the beginning has been modest and much work remains before the complete graphical communication problem is solved.

B. SCOPE OF PRESENT ACTIVITY

Sketchpad III is only capable of manipulating straight line, "wire frame," figures in three dimensional space.* Neither program writing nor a knowledge of computers is required to operate the system. The definition, construction, and manipulation of three-dimensional surfaces are not included at present; hence edges which are normally hidden by forward surfaces are not obscured as they should be. Since all edges are visible, one views a "wire frame" with no covering. Explicit information about the topology of the part is stored as it is sketched. Parts of an object (lines or endpoints) can be moved in space without erasing. All attached lines will follow the moving part.

C. MAN-COMPUTER INTERFACE

Real-time bilateral communication between man and computer is a prerequisite for a successful system. The CRT and light pen (described below) meet the hardware input-output requirements of fast, two-way operation.

1. Output: Visual Presentation

Graphical images of three dimensional objects are displayed on-line on a standard cathode ray tube. Because the screen is two-dimensional and the objects are three-dimensional wire frames, several viewing conventions were adopted to aid in visualizing the object in three-dimensional space. Stero-optic displays and similar methods of creating space sensations were not considered because of clumsiness and because of bilateral communication problems.

---

* Programmed for the TX-2 Computer, M.I.T. Lincoln Laboratory.

Four views of the object are displayed by program, one in each quadrant of the CRT screen.* A perspective view of the object appears in the

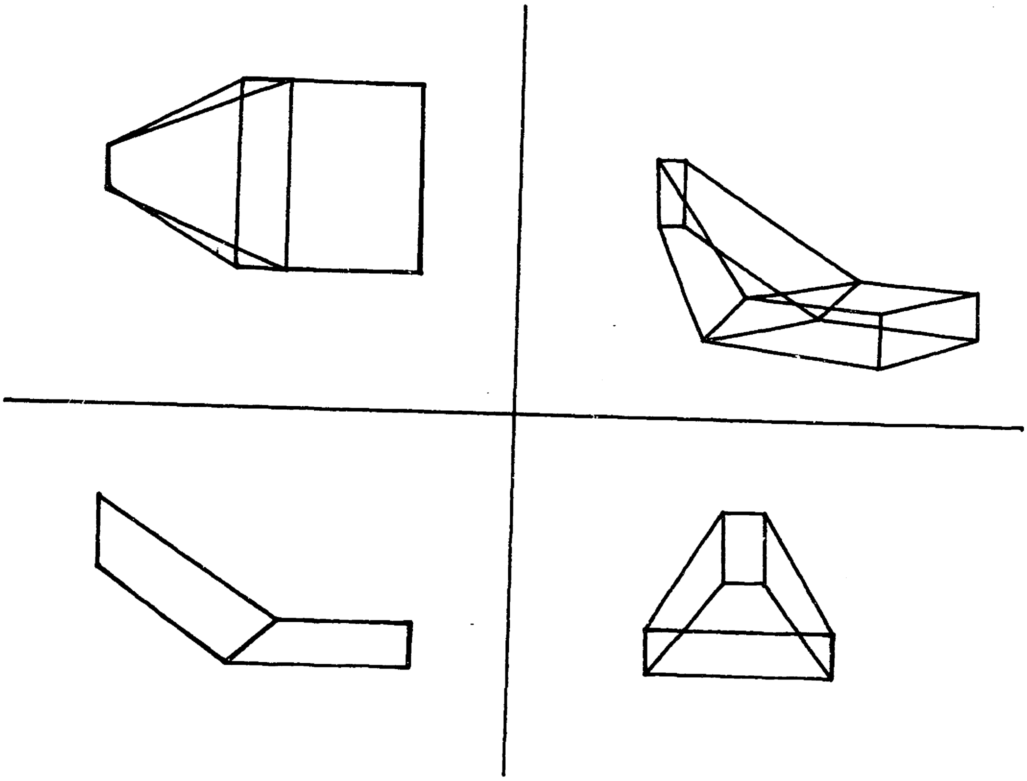


Fig. 1 Typical graphical presentation showing top, front, and side views plus a "3/4" perspective view.

upper right quadrant, and three orthogonal views in the remaining quadrants: top view--upper left, front view--lower left, and side view--lower right.

Wire frame objects displayed in a single-two-dimensional view without perspective fail to convey depth information. Perspective gives the illusion of three dimensions by supplying the familiar convergence of lines as they recede from the viewer. A single perspective view of an unfamiliar object does not convey visually all the correct information either; for

---

[*] All figures in this paper describing 3-D objects were drawn with Sketchpad III and plotted off line, by machine.

example, are the receding lines parallel or do they actually converge? Hence at least one other complementary view is necessary. Three projectively related orthogonal views were chosen for the complementary function. There are several reasons for this choice: a) The three views completely describe a straight line object in three dimensions. b) Three ninety-degree rotations of the part are simultaneously in view, reinforcing depth perception. c) Many users of Sketchpad III are uncomfortable sketching in perspective and would prefer the orthogonal system used by most draftsmen.

2. Input: Sketching

A light-pen is used to guide drawing on the face of the CRT. The light-pen is a photo-diode mounted in a pen-like housing which is connected to the computer. A lens system in the pen housing focuses light on the photo diode giving a field of view of approximately one-half inch when the pen is held within three inches of the CRT screen. The pen, which acts only as a receiver, responds if a scope phosphor is intensified within its field of view and interrupts the computer momentarily. Suitable programming determines which point causes the response. Thus, an existing line in a drawing can be singled out for program examination by merely pointing at the line with the light-pen.

The light-pen also controls the position of a light spot on the scope. This in turn controls the position of a three-dimensional point permitting the drawing of new lines. The program anticipates the two-dimensional position of the pen on the scope and displays a series of dots in the form of a cross at the guessed position. By noting which points in the cross fall outside the pen's field of view, the program calculates the position of a new cross that is closer to the center of the moving pen's viewing field. Thus the two-dimensional position of the pen is tracked by a light source. The light-pen permits the CRT screen to pass information in two directions; the CRT is simultaneously both an input and an output device.

Pushbuttons are provided which enable the operator to direct the computer program; for example, to erase what the light-pen is pointing at, to move what the light-pen is pointing at according to subsequent pen motions, to start drawing a straight line where the pen is pointing according to subsequent pen motions, to translate the drawing according to pen motions, and so forth.

The program interprets the rotation of three digital shaft encoders to mean: 1) magnify or reduce the drawing, 2) rotate the drawing clockwise or counterclockwise, 3) force or relax the perspective (by changing the position of the observer relative to the object. This causes the apparent convergence of parallel lines to change).

D. GRAPHICAL TRANSFORMATIONS

The four projections viewed on the scope are not four independent displays of stored two-dimensional information; rather, space coordinates in a single data structure are transformed into two-dimensional images for display. Rotating, translating, magnifying, and changing the perspective does not affect the data structure, (local transformations excluded). A line being drawn in any one view is simultaneously seen in the three other views; lines are in effect drawn in three-dimensions and simultaneously fed back for display.

Rotation, magnification, translation, and perspective transformations are performed by a single 4 x 4 matrix.* Two matrices are used for display purposes, one for the perspective view and the other for the three orthogonal views -- thereby enabling the perspective view to be manipulated independently of the orthogonal views.

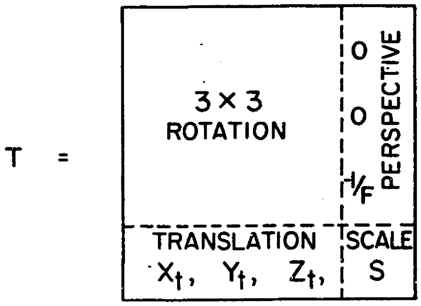
The matrix can be viewed as a partitioned matrix of four parts. The upper left is a 3 x 3 rotation matrix. The lower right position is the scale factor. The row at the bottom contains the three translation terms. The column on the right holds the inverses of the three viewing distances used in the perspective transformations.

The matrix operates on homogeneous coordinates (a, b, c, s). A three-dimensional point is determined by the ratio of its homogeneous coordinates:

$$a:b:c:s = x:y:z:1$$

---

* Developed by Lawrence E. Roberts at the Massachusetts Institute of Technology for use in his doctorate thesis on assembling three-dimensional descriptions of objects from their photographs.[2]



NOTE:
In this example, F equals the Z distance from the X-Y projection plane passing orthogonally through the origin.

(a) 4x4 Transformation Matrix Viewed as Four Partitioned Sections

$$\boxed{X, Y, Z, W} \times \boxed{T} = \boxed{X+W(X_t),\ Y+W(Y_t),\ Z+W(Z_t),\ (\tfrac{-Z}{F})+WS}$$

$$\downarrow$$

Converting homogeneous coordinates for display in the X′-Y′ plane gives:

$$X' = \frac{X+W(X_t)}{\frac{-Z}{F}+WS}$$

$$Y' = \frac{Y+W(Y_t)}{\frac{-Z}{F}+WS}$$

(b) Typical Transformation of Homogeneous Coordinates Using 'T' of Fig. 2 (a) as an Example (no rotation)

Fig. 2 Matrix configuration used to perform multiple graphical transformations.

In other words, the s coordinate is a scale factor and,

$$x = a/s$$

$$y = b/s$$

$$z = c/s$$

A typical transformation is shown in Fig. 2.

The rotation section of the matrix is changed by post-multiplying by a second rotation matrix. This second 3 x 3 matrix describes relative two-dimensional rotations about one of the three orthogonal axes of the scope screen.

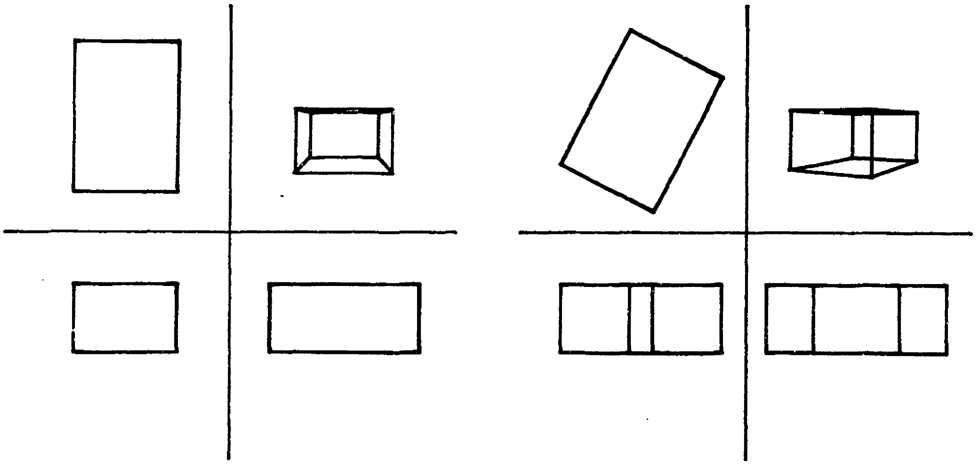
To cause a rotation, the operator first selects an axis of rotation by pointing at one of the three orthogonal views with the light-pen. Then the rotation shaft encoder is turned and the part rotates about an axis perpendicular to the selected viewing quadrant. The program calculates the sine and cosine of the shaft angle and positions the results in the second matrix according to the selection of axis. Post-multiplication then takes place and the second matrix is cleared in preparation for a new loading. The resulting new transformation is applied to the drawing and a new display is "painted" on the scope screen. The program continually samples the knob position; as long as the operator continues to turn the encoder, the process repeats. The cycle is fast enough in the case of simple drawings to give the illusion of a moving picture. The part moves relative to two of the three possible axes of rotation in three dimensions as the picture is rotated about one of the orthogonal scope axes. This method of rotating about one of the scope axes is visualized more easily than rotation about some axis fixed with respect to the part. Since the part can be rotated to any attitude in space by the twist of a knob, projectively related auxiliary views of the object can be generated at once. (Figure 3)

The remaining sections of the transformation matrix are manipulated in similar fashion. Application of the matrix to segments of the object singled out by the light-pen gives the important local transformation feature. Thus, graphical modification, which is so vital in the design process, is readily available.

E. DRAWING IN THREE-DIMENSIONS ON A TWO-DIMENSIONAL SURFACE

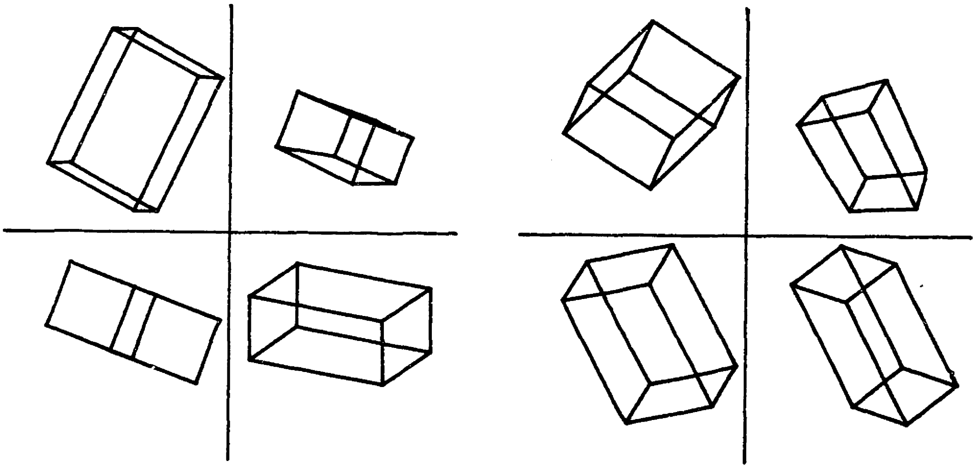
Moving a light-pen across the face of a CRT and having the correct three-dimensional information pass into computer memory requires a simple method of specifying depth coordinates. Several methods were considered, such as assigning the depth coordinate to the pen location by positioning a shaft encoder or joystick. Approaches like this were dismissed because of the nearly "artistic" talent an operator had to have to visualize movements of the point.

The method finally adopted utilized the rotation facility. Because the part can be rotated to any position in space, lines can be drawn directly in three dimensions by drawing in a plane. The part is rotated until the area in



A) Viewed Orthogonally

B) Rotated about 2nd quadrant axis



C) Rotated about 3rd quadrant axis

D) Rotated about 4th quadrant axis

Fig. 3 Additive rotation, beginning at (a), of a "wire-frame" box about axes perpendicular to the orthogonal viewing quadrants. The first quadrant in each subfigure shows a perspective view of the third quadrant, or front view.

which the line is to be drawn is parallel to a viewing quadrant. The line is then drawn true length; the depth coordinate remains constant as the pen moves across the plane of the scope screen. Specification of the single depth coordinate is done by program interpretation as described below. This program, called the Pen Space Location program, is the backbone program of three-dimensional sketching. Sophisticated drawing is made possible by this program. Because of its generality, rotation of the part to bring sections into true view is not always necessary.

F. PEN SPACE LOCATION PROGRAM

The Pen Space Location program performs two important functions: a) it defines a point in space called the Pen Space Location (PSL), by assigning a depth coordinate to the pen location according to the pen's previous position in another view, and b) it permits precise positioning of the PSL with coarse light-pen motion. It is the motion of the PSL that guides the drawing of lines.

1. Arbitrary Depth Assignment

The three orthogonal views represent three adjacent faces of an imaginary cube surrounding a part in space; the edges of the part are projected onto each face. A point on any face defines a projection line in space perpendicular to the selected face.

When defining a three-dimensional point, the Pen Space Location program begins the definition by interpreting the projection line of a two-dimensional point positioned by the pen as the intersection of two temporary, invisible planes parallel to the faces of the imaginary cube. To complete the definition of the point in space, a second point is positioned in either of the other two views and its projection line will then intersect one of the two temporary planes. This intersection defines the point in space. (Figure 4)

The two-dimensional position of the light-pen on the scope face is established by the tracking cross program. This program calculates the viewing field center coordinates of the pen and remembers the pen position when the pen is removed from the scope face.

When the pen begins tracking in a second view, the Pen Space Location program obtains the appropriate coordinate from the remembered point in the first view for use as the depth coordinate of the current pen position. This is the pen position in space called the Pen Space Location and its projections are displayed simultaneously as bright dots in all the viewing quadrants. As the pen is moved in one view, the depth coordinate of the PSL projected in front of the pen remains fixed.

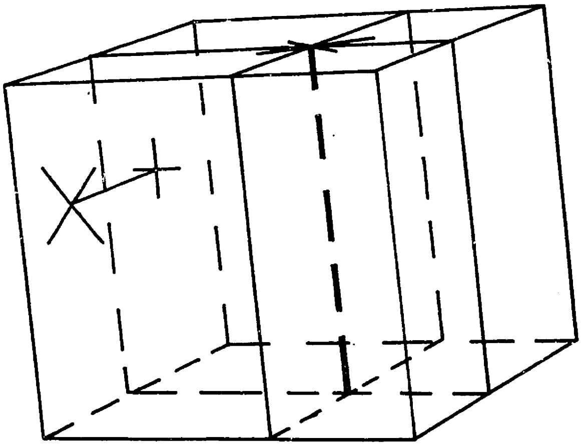


Fig. 4 Graphical interpretation of method used in the PSL program to define a 3-D point. The cube represents the imaginary volume enclosed by the orthogonal viewing quadrants. In this example, the double dashed line indicates the projection of the first point positioned in the top view. (Dashed lines are used for visualization purposes only.) This line is interpreted as the intersection of two planes. The small "X" represents the second point positioned in another view. The solid line radiating from the "X" is its projection. The "+" defines the 3-D point.

Lines are drawn by locating the PSL at an initial position, depressing a Start Draw button, and moving the pen. As long as the pen tracks, a line appears between the initial point and the PSL. Removing the pen from the screen with a flick of the wrist terminates the line at the last position of the PSL.

2. Precision Point Assignment

To begin a line on an existing line, the PSL must be precisely positioned on the desired section of line. Since the light-pen motions directing the movement of the PSL through space are quite coarse, programming must direct the precision movements of the PSL.

The PSL is surrounded by an imaginary sphere of one-eighth inch radius. The coordinates of the PSL and the radius of the sphere are converted to homogeneous coordinates and transformed by the inverse of the 4 x 4 transformation matrix. When the pen is held over a line, the PSL program determines which line (or lines) is responsible for the light-pen response and compares the stored line(s) with the transformed PSL. If a line passes through the sphere surrounding the PSL, the program calculates the point on the line nearest the center of the sphere and assigns the PSL this new value. As soon as the line passes outside this sphere (centered in front of the pen), the PSL assumes a value that is projectively related to the viewing field center of the pen. The new PSL retains the depth coordinate of the point on the line. The Pen Space Location program also determines end points and intersections of lines in this manner. Thus a steady hand is not required to construct accurate drawings.

The PSL program has a second mode that can be selected by push button. Instead of surrounding the PSL with a sphere, the projection of the PSL is surrounded by a cylinder. Whenever a line intersects the imaginary cylinder, the point on the line nearest the axis of the cylinder replaces the PSL. As in the other mode, when the lines fall outside the detection volume, the new PSL retains the depth coordinate of the point on the line, and is projected in front of the pen. The cylinder mode is generally used:

a. for selection of existing depth coordinates in one view. That is, it is not necessary to move the pen momentarily to a second view to establish a depth provided a line already exists there, and drawing can proceed at a faster speed. This is particularly useful when drawing in the perspective view.

b. to determine the location of a projected line in three dimensions. The PSL is "locked" onto the line by moving the pen over the line in one view and the projection of the PSL can be observed in the other views.

The section of the PSL program that performs nearest point calculations is flow diagrammed in Fig. 5. The cylinder mode is merely the special case where the depth coordinate is ignored. The mode selection button modifies the program so the proper third coordinate is ignored.

G. SUPERPOSITION

The cylinder mode fails when lines are superimposed. Attempting to "latch" the light-pen onto a line projected on top of another line confuses the program and the nearest point calculations are performed on both the lines

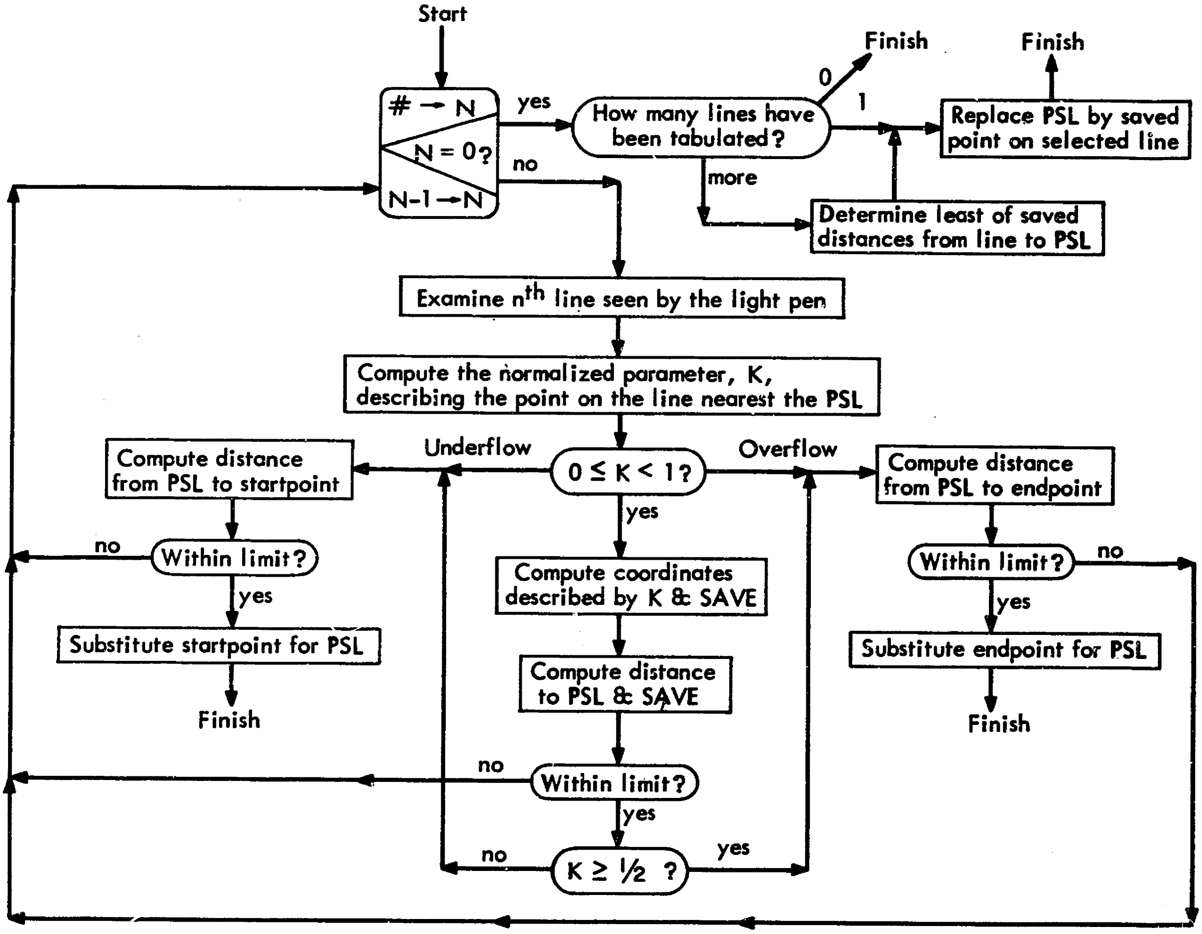


Fig. 5 Flow diagram of the precision point assignment section of the PSL program.

"seen" by the light-pen. If a cube were rotated so it appeared as a square, for example, the operator could not predict what depth the PSL would assume when the pen is held over a line. Thus, when the superposition occurs, the sphere mode must be used.

H. SYNOPSIS OF SKETCHPAD III UTILITY PROGRAMS AND DATA STRUCTURE

The data structure and the utility programs that generate the data structure used in Sketchpad III were developed by Ivan E. Sutherland at Massachusetts Institute of Technology for use in his two-dimensional graphical communications program, Sketchpad. A general introduction to the utility programs is given below, but the reader is referred to Reference 3 (in the Bibliography) for complete details.

Graphical information is stored in an n-component list structure. The pointers connecting the n-component elements form closed rings and enable tracing of information in either direction. Different rings thread through several levels in an n-component element and thus provide several paths to the same information. Each type of graphical element such as a line or a point, is referenced by a single block. These blocks contain pointers to the appropriate transformation and display subroutines. Each of these distribution blocks in turn is grouped according to generic type.

The program that operates on the structure is generalized. An executive routine determines what type of graphical manipulation must be performed by periodically sampling knobs and pushbuttons. The graphical element to be manipulated, indicated by the light-pen or pushbutton is found in the list structure; the program subsequently transfers to its distribution block via the list structure. The distribution block in turn transfers control to the subroutine which determines how the manipulation is done. Thus, the program is composed of many modules which are interconnected by the list structure itself.

Many features have been adapted from Mr. Sutherland's program to increase the flexibility of Sketchpad III. These include:

1. Storing several pictures in computer memory, subject to immediate recall.
2. Storing pictures on magnetic tape.*
3. Merging or combining end points of lines.
4. Merging two lines into one line.
5. Generating hard copy on an X-Y plotter.*

A program, now almost completed, inserts one picture into another picture. This feature accelerates the drawing of repetitive structures and the integrating of components into assemblies.

---

* Program by Leonard M. Hantman of Lincoln Laboratory.

I. FUTURE ACTIVITIES

Many difficult problems must be surmounted before a general graphical system is operating. Methods fast enough to satisfy real-time requirements must be devised that:

1. Define arbitrary surfaces.
2. Determine space-curve intersections of two surfaces.
3. Determine edges hidden by arbitrary surfaces.
4. Satisfy general graphical constraints.

As the system evolves, computers will be applied throughout the design-to-manufacturing spectrum. Design analysis (stress, dynamic, etc.) capabilities will be embraced by the system and operate directly upon and influence the stored graphical information. Manual part programming for numerically controlled production machines can be by-passed. The goal is to decrease the time spent preparing a part for production (lay-out, detail, drafting, etc.) from months to days.

CHAPTER II

HISTORY OF SKETCHPAD III

The method used in Sketchpad III to draw directly in three dimensions is simple and straightforward. However, this deceptively simple approach was not immediately evident at the outset of Sketchpad III development -- as has often been said, invention is the process of discovering the obvious.

Research began by attempting to throw out the concept of pen and paper (this proved to be the most difficult step). Realizing the digital computer would allow a new type of graphical communication, the problem lay in discovering it. The new communication method had to be general, simple to use, and take advantage of the logical power offered by the digital computer. The display and input medium had to be an integral part of the communication system. Thought was first directed toward the less abstract problem of input-output.

A. HARDWARE SEARCH

At that time, September 1961, the light-pen was available, but considerable skepticism existed about its use as a precision drawing device. A project spokesman allegedly noted, "Drawing with a light-pen is like trying to write your name on a wall fifty feet away with a pistol."

Many ideas circulated such as: 1) attaching a manually directed stylus to a pantograph which drives two shaft encoders. The two-dimensional position of the pen would be determined by sampling the encoders, and 2) directing the movements of an X-Y plotter by a manual joystick device. These devices were unsound for several reasons: a) erasing would be difficult, b) presentation would be slow and clumsy, and c) the pen and paper concept was not escaped.

In October, 1961, Ivan E. Sutherland demonstrated the power of the light-pen as an accurate drawing device while developing his thesis. The oscilloscope became a candidate for the display of three-dimensional information; erasure and speed problems evaporated. But how could three-dimensional information be best displayed on the two-dimensional oscilloscope screen? Perhaps direct three-dimensional presentations were needed. Two oscilloscopes could generate stereo-optic displays, but the idea was reluctantly discarded, because of the clumsiness of eye-glasses or other means of image separation.

One industrial lab had just completed a new prototype three-dimensional display system. The principle used is simple: a transluscent screen is rotated about a single axis fast enough to become "invisible." A projection oscilloscope repeatedly strobes the screen every time the surface moves into a desired position. The effect is a point of light in three-dimensional space. Generating thousands of points in this fashion developes a ghost-like image in space. The display is simple in principle, but difficult to implement because of electronic complexity.

Seeing that the technology of direct three-dimensional display was inadequate, the necessity of direct display was questioned. Depth perception is aided visually by three independent means: 1) the binocular effect, 2) the perspective effect, and 3) the obstruction of rearward surfaces by opaque, forward high-lighted surfaces. The independent nature of these aids is demonstrated every time an object over thirty feet away is studied. Binocular vision fails at this distance and perspective is the next aid we unconsciously call upon. Perspective fails, also, if the object is totally unfamiliar. For example, unless previous experience has taught us that the object in question has vertical or horizontal edges and is a certain size, one cannot determine the relation of the object to its surroundings by perspective alone. If we are able to move relative to the object, or if it is moving relative to us, we call upon the third aid. We observe rearward surfaces rotating toward us and hiding the receding surfaces. Light plays upon the surfaces varying the shading and the high-lighting. The bulk of depth information we receive is, in fact, gleaned from the last two aids: perspective and the opaque nature of surfaces coupled with lighting effects (using previous experience, of course). One does not normally have a difficult time determining depth relations when viewing a movie cartoon which uses little of the shading effect. Thus, binocular or stereo effects appear to be unnecessary for depth perception.

The two dimensional CRT screen could duplicate perspective and shading. However, shading effects due to simulated lighting and simulated surface texture are difficult to produce by computational methods, so this visualization aid was dismissed. The question then became: would perspective and surface interaction through obstruction alone provide the observer with sufficient sense of depth to permit him to visualize ordinary mechanical parts? Although the answer is not clear even today for parts containing arbitrary surfaces, it was felt then that given the ability to move the part in perspective, so that relative motions of edges could be detected, the observer would have suitable information to establish the desired three-dimensional perception.

B. PROBLEM AREA DEFINITION

With the display question settled the next problem was to narrow the area of investigation. No one had studied this field before, so work had to start from scratch. Surveying the expanse of general graphical communication problems, such as arbitrary surface manipulation, was an overwhelming experience. Since so much had to be learned, the least complex problem was chosen: graphical description of straight-edged objects -- in hopes that the experience gained in this restricted area would indicate a way to a solution of the general problem. The term straight-edged object means that the intersections of all surfaces are straight lines, although the object does not necessarily have plane surfaces.

Straight-edge objects present a relatively simple problem because straight lines remain lines after any graphical transformations (unlike transforming a circle, for instance) and the display of straight lines on existing hardware is simple and fast.

Surface and hidden line determination are excluded from this thesis, because of lower priority. However, this area is so important for visualization that already it is being considered. For convex objects[*] hidden line

---

[*] An object in which any two points on or within the volume can be connected by a straight line that lies wholly within the volume.

determination is simple. But for the general class of objects containing concavities, such as an angle iron, or a series of separate objects that are not connected together, the problem remains a difficult one because of real-time communication requirements -- the ability to move the part in a continuous fashion. Ideally a new incremental motion should be displayed every thirtieth of a second on the CRT screen (thirty frames a second) to give the illusion of flicker free motion. Hidden edge routines for concave and separated objects being developed today use search techniques (see Reference 2 in Bibliography). On today's machines search techniques are too slow to satisfy the real-time requirement.

C. DISPLAY PRESENTATION CHOICE

As soon as the problem area was defined, thoughts turned to display techniques. Only a perspective view was available for depth perception unless another view was displayed simultaneously. Three additional views were added. The orthogonal views were added for more complex reasons than simply the designer having difficulty drawing in perspective.

A curve in perspective will appear smaller as it is moved farther behind the projection plane. Orthogonal projections exhibit true shape and size of any plane curve parallel to the projection plane regardless of depth. This property of the orthogonal projection system is important in mechanical design where exact spatial relations must be rapidly communicated and manipulated by the designer. Given three orthogonal views, no conjecture is necessary to determine the relation of one line to another. The orthogonal relation of the three views is a consequence of the projection lines. The projection lines are imaginary rays passing perpendicularly through a viewing plane and intersecting the edges of the part. When more than one viewing plane is required, additional planes are oriented orthogonally, so the projection lines of the first view appear in true length in the remaining views -- again, an important property for use in graphically communicating and manipulating true shapes and sizes.

D. INITIAL SYSTEM INVESTIGATIONS

Although the use of orthogonal views is sound, the choice proved misleading at the outset of the project because their use was associated with descriptive geometry. The orthogonal projection system is the familiar tool of descriptive geometry -- the field of graphics describing a three-dimensional object on a two-dimensional surface. Because of years of experience using descriptive geometry, work began with pen and paper methods in mind. The approach was to draw two views and store each view as related two-dimensional information. The program was to compile this information into a three-dimensional description. Compiling was thought advantageous at first. For example, a cube could be drawn using only eight lines -- four describing a square in the top view and four for a front view square. The program would compile the description into the twelve lines representing a cube. The designer was relieved of describing every line in space. However, graphical ambiguities were easily generated. For example, a cube described as a square in the top and front views could just as easily represent a wedge. To overcome this, the program was written to guess an interpretation and display it; if it was incorrect, the designer would alter the interpretation. Unfortunately, the time saved in compiling an interpretation could be easily lost in correcting the interpretation of the computer. A further drawback of the system was the storage of three-dimensional information in the form of closely related two-dimensional information. To relate points in the top and front views projectively, the data was stored in two list structures with common lateral X coordinate blocks linking the two structures. Thus, all lines terminating at the same X position in any view were related. This was to aid the compiling program, but it meant rewriting the list structure every time a rotation was made -- a time wasting procedure. Three months of programming the M.I.T. TX-0 computer elapsed before disenchantment set in. Abandonment occurred when the compiler-drawing program began to operate and one could experiment by actually drawing. The time spent programming the computer was, however, not wasted. Experience gained in programming for real-time operation and composing list structures allowed graduation to a new phase of investigation.

In retrospect, it was learned that graphical information may be introduced into computer memory by drawing two related two-dimensional views and may even be stored two dimensionally in coordinate pairs, but no attempt should be made to share coordinates with elements other than the common points of intersection.

E. THE DAWNING OF SKETCHPAD III

Late in August, 1962, M.I.T. Lincoln Laboratory extended an invitation to use their TX-2 computer. The TX-2 computer was specifically designed for rapid input-output in the real-time domain (See Appendix A) -- a prerequisite for successful graphical communication.

At this time, exposure to Lawrence Robert's multiple transformation matrix occurred. His compact, unified approach to transformations sparked new thought and the present approach of drawing directly in three dimensions was born.

Ivan Sutherland's Sketchpad programmed for the TX-2 computer was in operation at that time. The move to the TX-2 enabled the incorporation of many of Mr. Sutherland's utility programs into the Sketchpad III system, saving vast amounts of programming and debugging time.

During the period of revision, while Sketchpad III was forming, certain reservations arose over the present restriction of drawing in a plane, even though the plane orientation was arbitrary (the depth coordinate remains fixed as the light-pen moves in one view). When only the endpoints of lines are to be positioned, the restriction need not apply because the viewing quadrant can be rotated or the pen can be moved to another view during drawing. The concern arose because of anticipated future expansion into surfaces: could arbitrary surfaces be defined with planar curves? Professor Steven Coons, co-supervisor of the Computer-Aided Design project, quickly evaporated all reservations. He pointed out several economical methods where arbitrary surfaces could be constructed with plane curves (to be documented in a forthcoming project report). Armed with this reassurance, development of Sketchpad III continued in a more confident vein.

Even at this late date, pen and paper methods were still an influence. The idea of operating the light-pen in a spherical mode had not yet occurred and a mode similar to the present cylinder mode was being considered.

Superposition was a problem. How was the light-pen to know which of many coincidently projected lines was to be singled out? First thoughts ran to pointing at the unwanted (or wanted) lines in other views and indicating by pushbutton that they were to be made invisible to the light-pen in the offending view. This was a poor solution, for the operator could labor minutes trying to locate all the lines. Eventually the obvious dawned and Sketchpad III took its present form.

CHAPTER III

GRAPHICAL TRANSFORMATIONS

A. MAGNIFICATION

The eight inch square viewable area of the TX-2 oscilloscope is composed of 1024 x 1024 discrete points (ten bits describe each coordinate). Magnification of the display is available since the TX-2 word size (36 bits) is larger than the two 10-bit fields decoded for display. A magnification of 2^8 (256 powers) is possible by storing the two coordinates as two 18-bit words in one TX-2 word. However, this is not done because of the nature of homogeneous coordinates and because the TX-2 is restricted to fixed-point arithmetic. Each homogeneous coordinate, including the scale factor, occupies a 36-bit word. Regarding the coordinates as fractions, the coordinate sets (x, y, z, w) are normalized so the largest coordinate does not equal or exceed $|\frac{1}{4}|$ for overflow reasons. The significant digit of the smallest coordinate is not allowed to drop below $1/2^{11}$ so as to prevent underflow. This gives a magnification range of 2^{18} or 262,144 powers. This figure is used out of curiosity and represents an extravagantly large volume. For example, the sphere of the earth could be displayed with enough magnification available to resolve objects down to 160 feet in diameter. Clearly, no memory is large enough to store all the graphical information that would occupy this large volume. Nevertheless, large magnifications are useful when accurate graphical positioning is desirable, such as describing precision parts for fabrication by numerically controlled machines. The program interprets changes in the scale shaft encoder as exponential (rate of change is proportional to drawing size) increments which produces rapid changes in the transformation matrix scale factor.

B. TRANSLATION

Any of the four CRT displays are translated by light-pen and pushbutton. The pen is moved to a desired view and the proper pushbutton is depressed. This cements the entire view to the pen and subsequent movement of the pen translates the drawing in the two dimensions of the selected viewing plane.

The program samples positions of the tracking cross, computing the two incremental movements in scope coordinates. These values are multiplied by the matrix scale factor giving movements in fractions of transformed oscilloscope diameters, which in turn are added to the appropriate positions in the translation section of the transformation matrix (Fig. 2). The endpoints of the object are transformed by the new matrix for display and the cycle continues until tracking is terminated with the customary flick of the wrist.

C. PERSPECTIVE

1. View Selections

A separate transformation matrix is used to generate the perspective view, enabling independent manipulation of this view. A perspective of the top, front, or side view can be selected by pushbutton. All movements -- rotation, magnification, or translation -- occurring in the orthogonal views are seen in the perspective view (Fig. 3). A fourth choice is the pushbutton selection of a permanent $\frac{3}{4}$ perspective view (Fig. 1) that does not move when the orthogonal views are moved. Thus, a standard orientation is always available for reference purposes.

2. Rotation

Either the slave or permanent perspective view can be moved independently of the master view. The view can be rotated by pointing at the first quadrant and turning the rotation shaft encoder. Here, the part is not rotated about an axis perpendicular to the scope screen, but about an axis perpendicular to the imaginary floor of the perspective view. This axis was chosen because motion is more easily visualized about this axis and all sides of the part can be rotated toward the viewer. (In all the viewing quadrants, the axis of rotation always passes through the center of the quadrant.) This dynamic examination of the part in perspective from all sides provides an invaluable visualization aid. Even small rotations of 10 degrees immediately clear up any depth confusion created by a static view.

3. Magnification

The perspective view is assigned its own scale shaft encoder, permitting an object to be simultaneously displayed at two different magnifications.

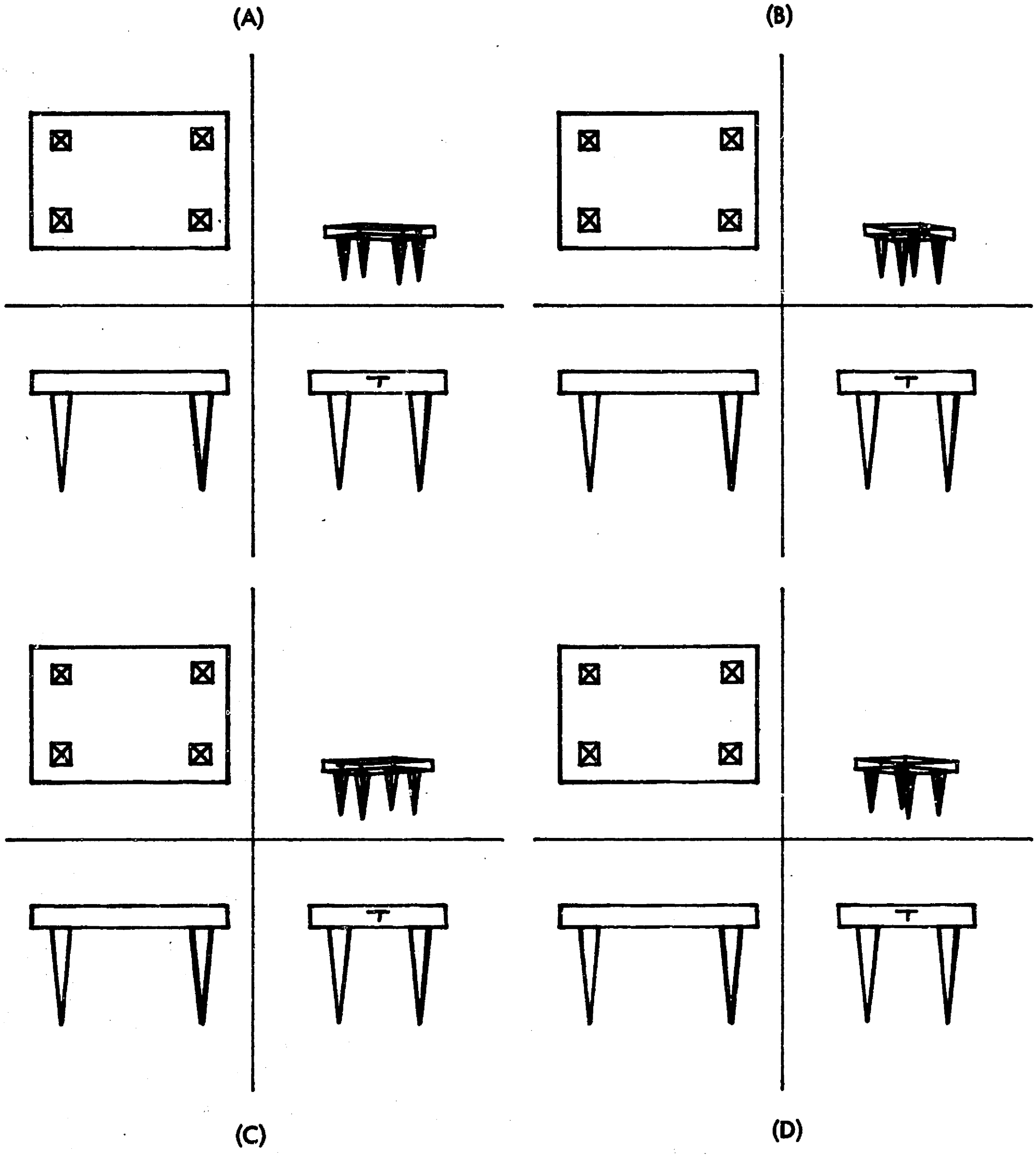


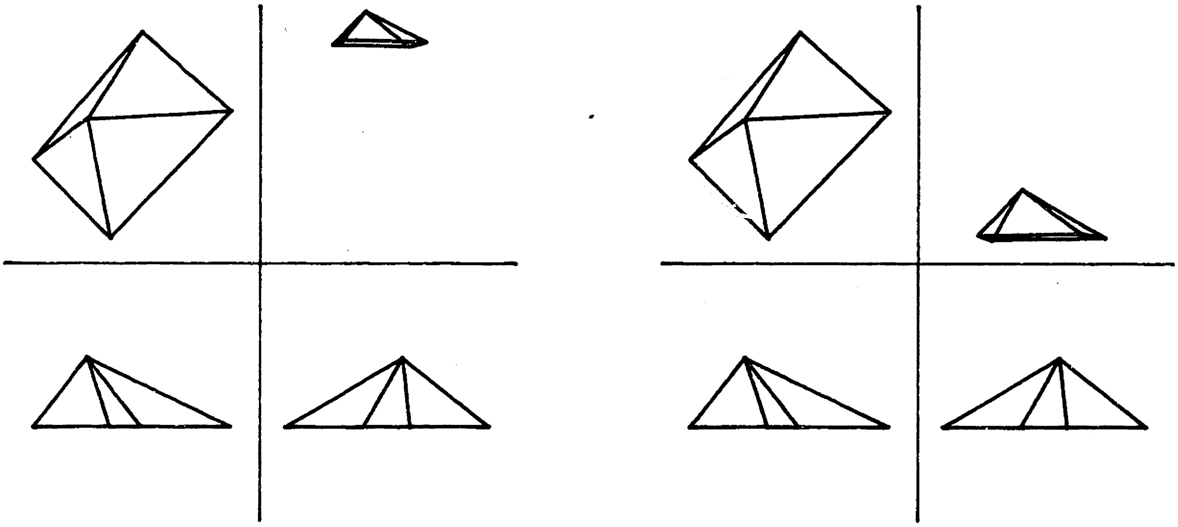
Fig. 6 Sequential rotation, beginning at (a), of the perspective view. The "T" on the end of the table is inserted to remove symmetry.

This is very useful when an overall view of the part must be maintained during detailed work. A second use occurs when a segment of an object is magnified in the orthogonal views so a line appears only in one view. Unable to determine the depth of the line in question from the orthogonal views, the part is reduced in the perspective until the entire object is in view. The light-pen is held over the desired line in the orthogonal view, cylindrical mode is selected and the PSL locates on the line. A glance at the perspective view tells where the line is in space.

Magnifying the perspective raises an interesting question: are we enlarging a two-dimensional projection or are we moving closer to the part (which changes the perspective)? Both operations are useful, so pushbutton control to yield both modes has been programmed.

4. Horizon Adjustment

Translating the perspective view so one component of translation is parallel to the vertical edge of the scope serves to raise or depress the part above or below the imaginary horizon line located at the center of the perspective viewing quadrant. This means the viewer is either looking at the underside or the topside of the part.



A) Viewed from Below

B) Viewed from Above

Fig. 7 Vertical translation of the perspective view. Vertical movements place the part above or below an imaginary horizon line passing through the center of the quadrant.

Figure 8 shows the graphical interpretation of the perspective transformation indicated in Fig. 2. The x-axis is the horizon line. The X-Y plane represents the viewing surface of the CRT, but could just as easily represent the film in a camera, since the same model demonstrates the operation of a camera lens system. E is the position of the eye from the

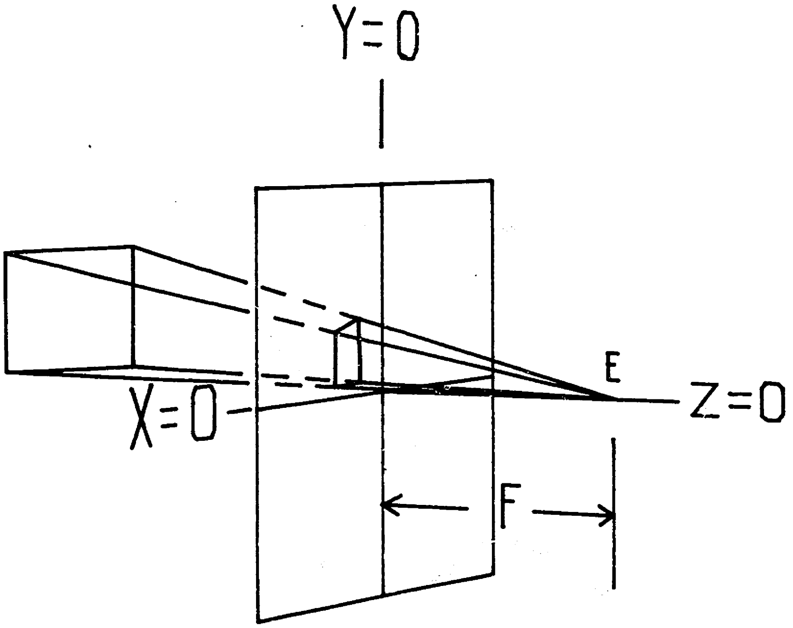


Fig. 8 Graphical interpretation of a perspective transformation. A plane figure is shown projected on the X-Y plane.

oscilloscope screen. As E moves closer to the screen, F (the perpendicular distance from the X-Y plane to E) decreases and compensation occurs which relates the display to the new distance of the eye. This compensation changes the rate of convergence of the receding lines or varies the perspective. Compensation is not automatic, but is accomplished by turning a shaft encoder. Variable perspective is also used to eliminate confusion about the relative depth of lines in crowded volumes by forcing the perspective. Note: increasing F to infinity changes the perspective view to an orthogonal view, because the projection lines become perpendicular to the X-Y plane.

Figure 2 shows the location of F in the matrix. Selecting a perspective view of one of the orthogonal views positions the current value of -1/F in the appropriate matrix position and automatically selects the two proper homogeneous coordinates for conversion.

D. TRANSFORMATION SPEED

The moving display effect depends on how fast an object description (endpoints) can be pushed through the transformation matrix. Transformation speed in Sketchpad III is realized at the expense of computer memory space. For space filling objects at least three lines must terminate at the same endpoint. If the display were formed by threading through the list structure to find the object's lines, and a line's endpoints were transformed as the lines were encountered, a single point would be transformed at least three times before the pass through the list structure was completed. To prevent this repeated transformation of endpoints, the program first transforms all the points in the object and temporarily stores the results within the list structure. Then the lines are located one by one, the related transformed endpoints are applied and a file of display parameters describing the objects' lines is calculated. Since a point is transformed only once, time consuming multiplication is minimized.

CHAPTER IV

PEN SPACE LOCATION

A. INTERPRETING LIGHT-PEN MOVEMENTS

When the light-pen is brought within range of the display, tracking is established at the first luminous point which falls within the field of view of the pen. The drawing is not displayed for approximately one millisecond, while the computer directs the display of the tracking cross. While the pen is within aiming distance new tracking crosses are intermittently displayed every 10 milliseconds to determine the new two-dimensional position of the pen. Thus, ninety percent of the time the central processing element is free from tracking chores and during this free time the drawing is displayed. During the main display the pen will interrupt the computer if it is held over a segment of the picture. The table in computer memory which holds the display parameters also contains in each parameter word the name (address) of the line represented by that parameter. Light-pen interruption allows the main frame to determine which display parameter caused the pen response before the next parameter is displayed. Once the display parameter is singled out, the program easily determines its associated line from the display table.

During a display cycle (called a frame), a small table of all lines which fall within the field of view of the pen is assembled. At the end of each frame, this assembly table contains all the lines that could have possibly been pointed at by the pen in one frame time. A second table is used to record a copy of the completed assembly table (completion signaled by the end of a frame) for use in the nearest point calculation program. During the next frame a new table is assembled replacing the assembly table for the previous frame. The TX-2 display is independent of the computational sequence; nearest point calculations are performed at the same time the display operations are taking place. Because of the serial nature of the display, in complicated drawings long times can elapse between the time when the pen sees a line and the time when the frame is completed. To insure that nearest point calculations keep up with the lines seen by the pen, any line being tabulated in the assembly table is inserted in the second table, if not already there. This speeds the latching process of the PSL, since a line will appear in the table operated upon by the nearest point program the instant it is seen by the pen.

B. DEMONSTRATING LANGUAGE

The error distance associated with the two detection volumes (sphere and cylinder) used to latch the PSL onto a picture part is smaller than the one-half inch field of view of the pen. (This allows more accurate and more selective pointing of the pen.) Once the pen is pointed at a picture element (a line or endpoint) and the PSL latches on, the operator can apply a graphical manipulation (erase, move, for example) to the indicated part by depressing the proper button. The program interprets the human's command as: "Apply function g to y," where y is the picture element and g is the graphical manipulation indicated by pushbutton. Thus, a single erase button, for example, serves for lines or endpoints. The man communicates with the computer by designating a graphical element to be referred to by a command.

C. NEAREST POINT CALCULATIONS

From the table of lines seen by the pen, the program examines lines sequentially to determine if the detection volume is pierced (Fig. 5). Since the homogeneous coordinates of the endpoints and PSL have floating scale factors, the point coordinates must be converted to new coordinates having equal scale factors before addition or subtraction can take place. If floating point hardware were available for the TX-2, this conversion would not be necessary.

Next, the normalized parameter, K, describing the point on the line nearest the PSL is computed. Normalized parametric representation of a three-dimensional point was chosen because the single, economical parameter is (a) sufficient to describe the closest endpoint of the line and (b) invarient in metric space.

Point (a) is important, because the PSL latches on to endpoints by using the topological relations of lines to their endpoints -- endpoints referred to the list structure need not be displayed. Endpoints are assigned a larger attraction force to the PSL than a line and must be mathematically examined before the nearest point on the line. The need to examine both endpoints is overcome by using K, since its magnitude indicates which endpoint is closest to the PSL.

Coordinate invariance in metric space is useful when more than one coordinate system is used during the computational analysis. When operating the PSL in cylinder mode, the coordinate system of the viewing quadrant is used in the initial stages of analysis and the coordinate system of the stored data is used in the terminal stages. K is invariant under rotation, translation, and magnification -- hence, coordinate conversion problems are overcome. Perspective transformations however <u>do</u> affect K, so conversion must be used when the pen is operated in the perspective view.

The derivation of K draws upon elementary analytical geometry.

If

$$\Delta x = x_2 - x_1$$

$$\Delta y = y_2 - y_1$$

$$\Delta z = z_2 - z_1$$

where subscript (2) indicates the endpoint of a line and subscript (1) indicates the startpoint, the equation of the line in parametric form becomes,

$$x = x_1 + \Delta x K$$

$$y = y_1 + \Delta y K$$

$$z = z_1 + \Delta z K$$

For the line segment between point 1 and point 2,

$$0 \leq K \leq +1$$

When K is greater than +1, a point is described on the extension of the line segment beyond the endpoint; similarly, $K < 0$ denotes a point beyond the startpoint.

To determine the coordinates of a point on the line nearest the PSL (x_p, y_p, z_p) in terms of K, one draws on the theorem:

> given the equation of a plane
>
> $AX + BY + CZ + D = 0$
>
> The coefficients of X, Y, and Z are the direction numbers of a line perpendicular to the plane.

One set of direction numbers is composed of Δx, Δy, and Δz. Therefore, the plane

$$(1) \qquad X\Delta x + Y\Delta y + Z\Delta z + D = 0$$

is perpendicular to the given line for all values of D. If the indicated plane passes through the PSL, the intersection of the line and the plane give the desired point.

D is determined from the condition that the plane passes through the PSL. Determining D by substituting x_p, y_p, z_p in (1) and rewriting (1) in terms of the known D.

$$(2) \qquad \Delta x\,(X - x_p) + \Delta y\,(Y - y_p) + \Delta z\,(Z - z_p) = 0$$

Substituting the parametric equation of the line in (2) gives the desired intersection in terms of K:

$$K = \frac{\Delta x\,(x_p - x_1) + \Delta y\,(y_p - y_1) + \Delta z\,(z_p - z_1)}{(\Delta x^2 + \Delta y^2 + \Delta z^2)}$$

Figure 5 shows that K is examined for underflow or overflow, which would indicate the nearest point is on the line extension. The magnitude of K determines which endpoint is closest to the PSL. The distance from this endpoint to the PSL is always computed and compared with the limit (the radius of the imaginary sphere or cylinder). Only when $0 \leq K < +1$ and the computed point on the line is within the distance limit is the line tabulated for later comparison with other lines that might fall within the limit.

D. PSL MODE ASSIGNMENT

The selection of spherical or cylindrical operation of the PSL requires a unique type of pushbutton for safety of operation. A drawing operation in three dimensions is unsafe if the PSL assumes a new depth when the operator does not want a depth change. If a mistaken depth were established -- perhaps by the operator accidently brushing the pen in cylinder mode over a line -- the operator would begin to construct a misshapen object.

When using spherical mode the chance is slim of establishing a new depth by passing over another line slightly removed from the drawing plane. Although a new depth can be established in spherical mode by following any line (latching on to a line and moving the pen along the line), the speed of this movement is slow, so an accidental brush with a line would not normally cause any change. Thus, the pushbutton controlling the PSL mode is spring loaded to remain normally in spherical mode -- the safe mode where an accidental brushing of the pen over a line does not change the depth of the PSL. Even though cylindrical mode is used more often, it is in operation only as long as the button is depressed.

Figure 9 shows a time exposure of the path assumed by the PSL in cylinder mode as the pen is brushed over lines of different depth.

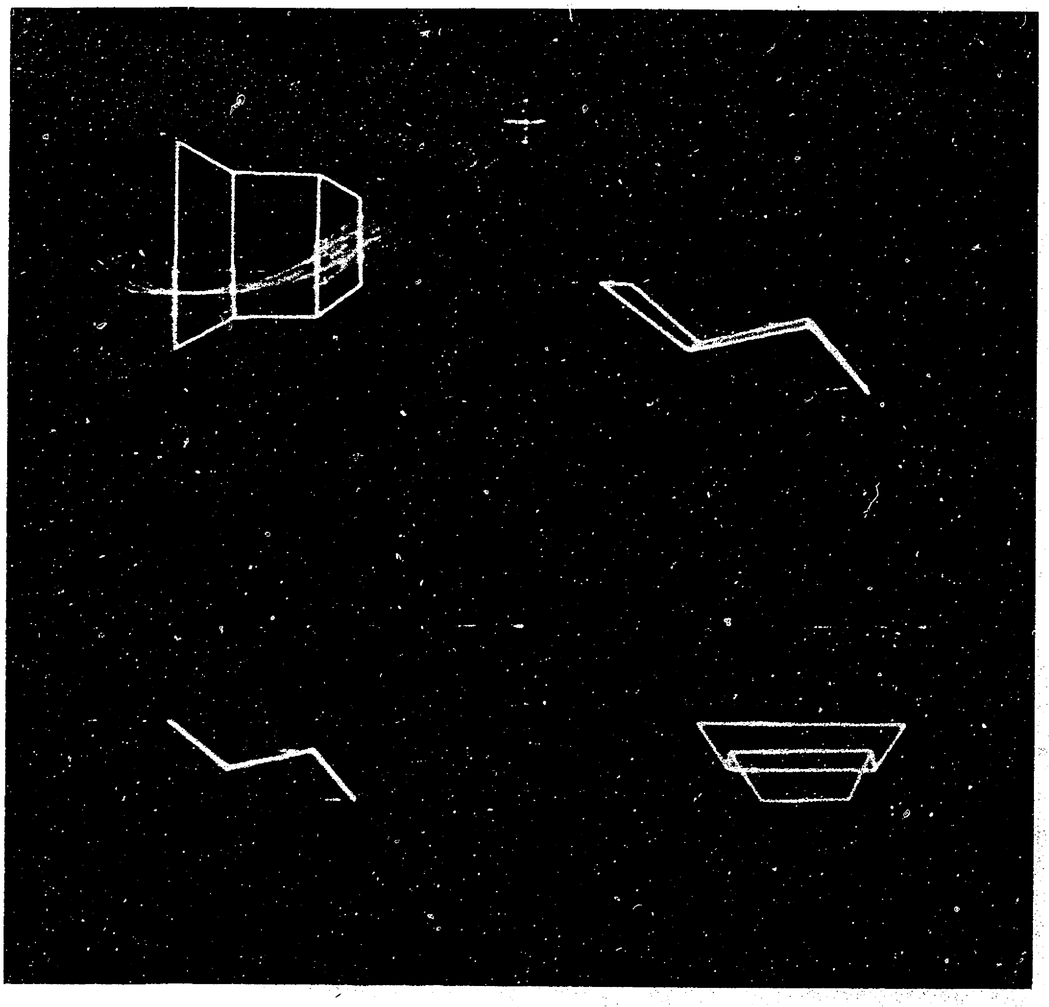


Fig. 9 Cylinder mode operation of the light-pen. Time exposure shows tracking-cross (light-pen position) moving in the top view (upper left quadrant) from right to left. The light, serrated lines show the path of the pen-directed PSL as lines of different depths are encountered by the pen.

CHAPTER V

EXAMPLES AND CONCLUSIONS

A. EXAMPLES

The examples in this chapter show how Sketchpad III could be used with its limited capabilities. Several illustrations indicate specific features of the system.

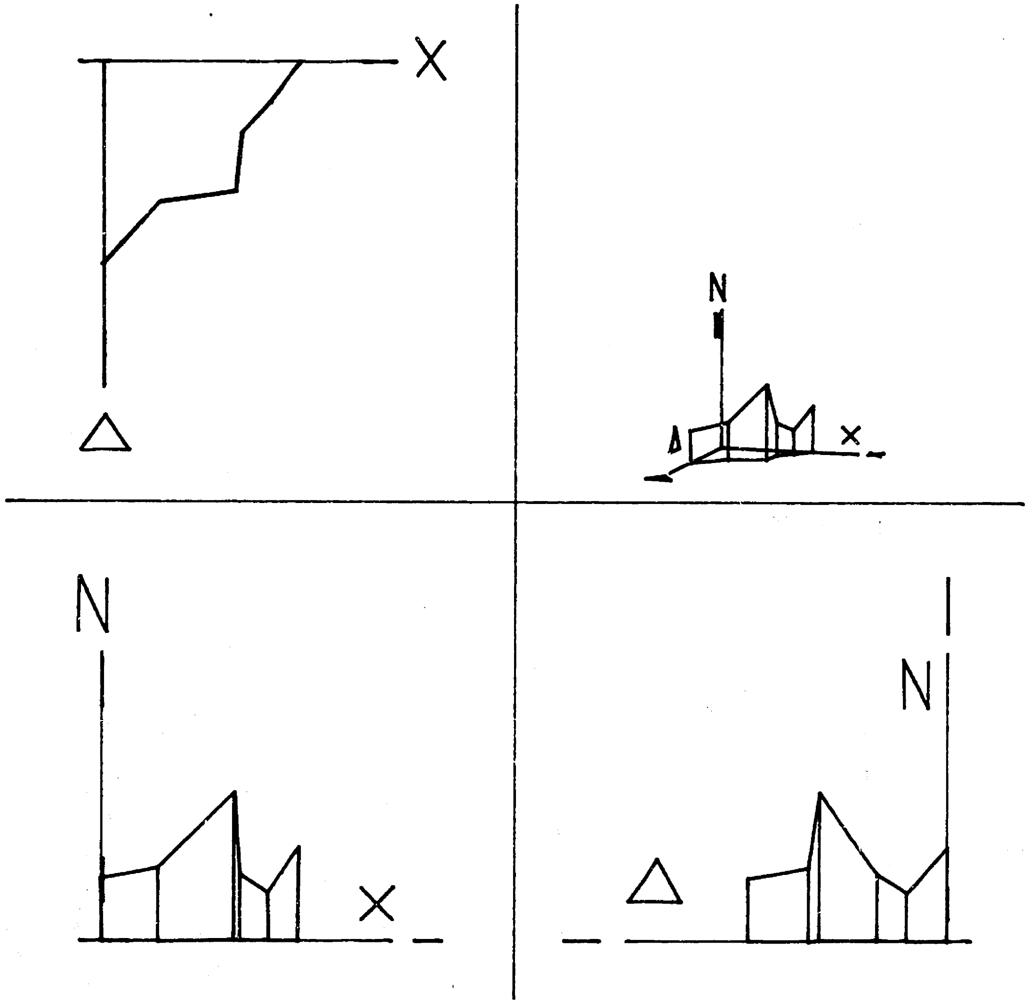


Fig. 10 Point-Line Graph

Three-dimensional point-line graphs have previously been static presentations where the views chosen for plotting are the only views available. Using Sketchpad III, one is not required to interpolate data, for any presentation can be displayed at the twist of a knob. Once a desired orientation is chosen, the results can be plotted off line for leisurely study.

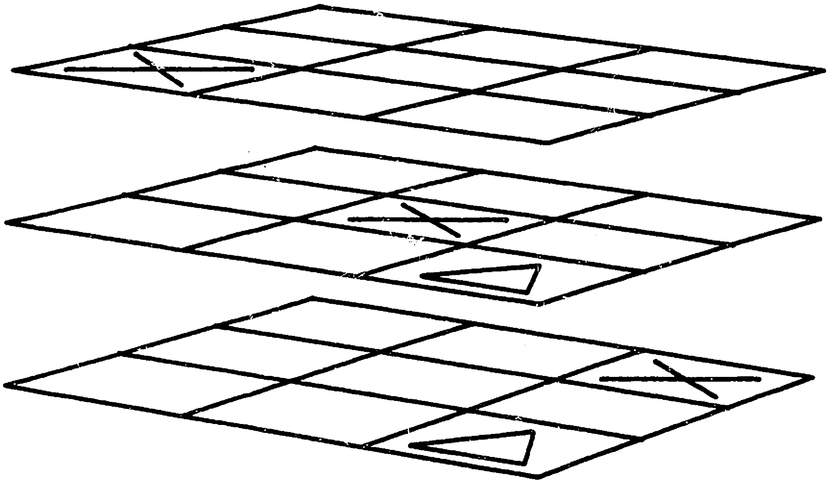


Fig. 11 Three-Dimensional Tic-tac-toe

Although a somewhat frivolous application, the three-dimensional tic-tac-toe game illustrates one of the dynamic qualities of the system -- erasure. Surely the experienced player of conventional tic-tac-toe would be overwhelmed at the extension into the third dimension and would frequently use the ERASE button ! Several hundred games could be stored in TX-2 memory.

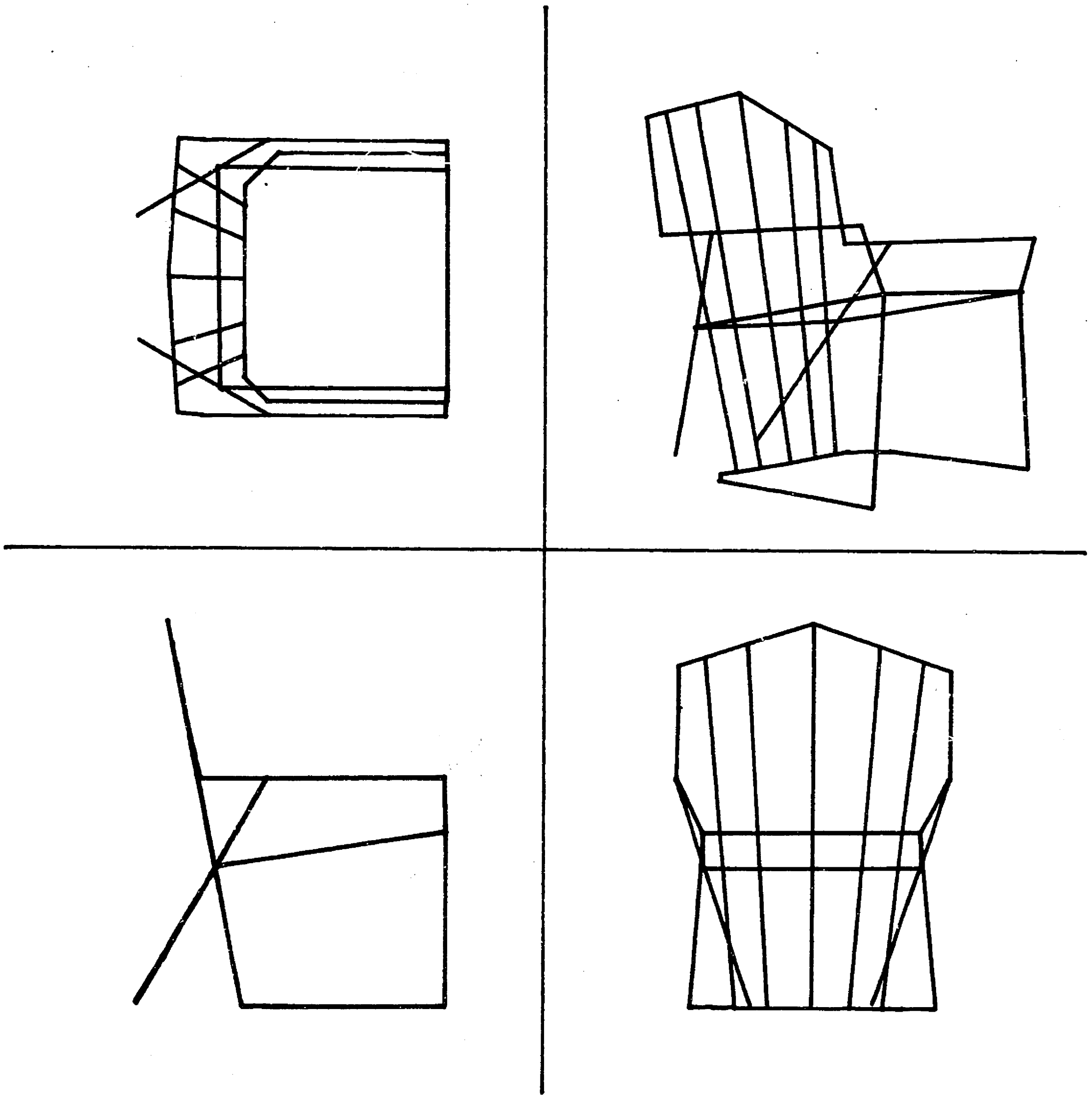


Fig. 12 Wrouaht Iron Chair Design

Figure 12, a wrought iron chair, shows the end product of additional system dynamics. By moving lines and points (the line terminators) while constructing the chair, one can readily change the design. Tilting the back of the chair, or raising the back with a single light-pen movement produces entirely new artistic effects. This continuous changing of component relations models the design process.

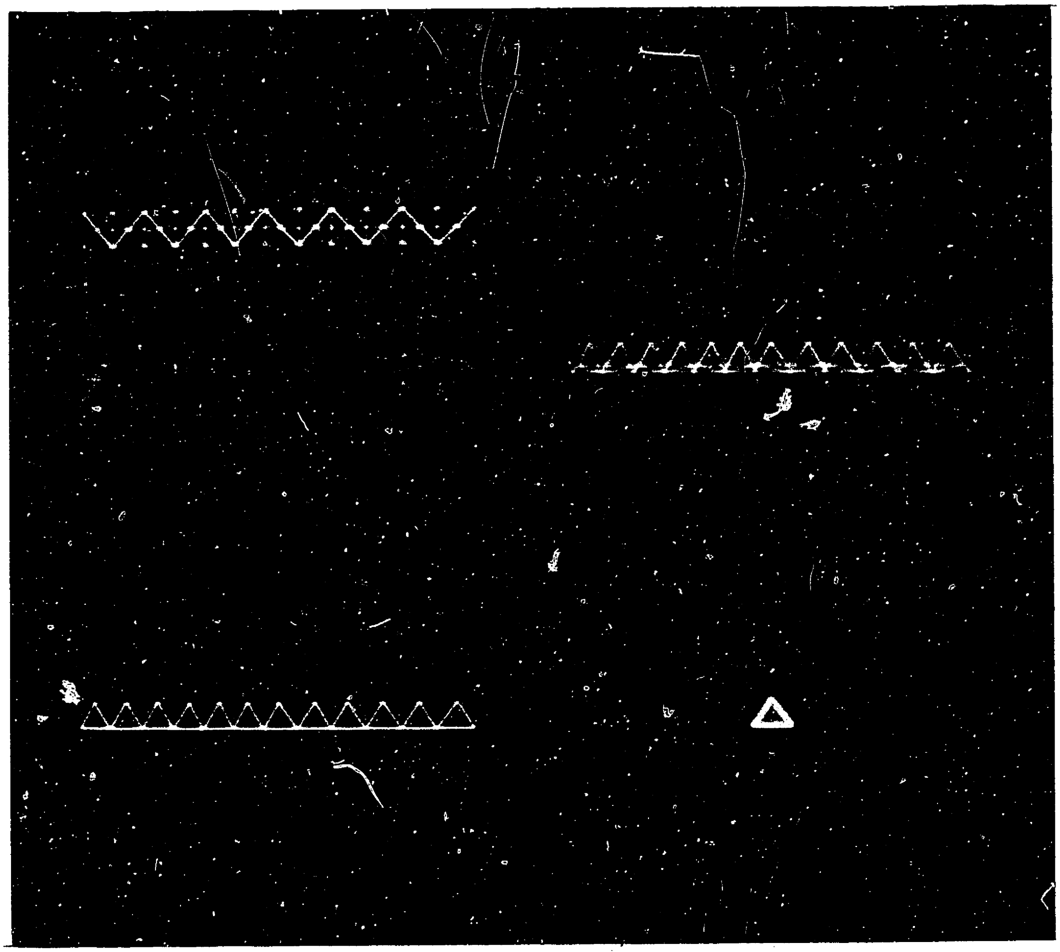


Fig. 13 Triangular Bridge Truss

The bridge truss shown in Fig. 13 was drawn for use as input to a stress analysis program. The part description in computer memory was written on magnetic tape for subsequent analysis on a different computer by a program written by Richard P. Parmelee of the Computer-Aided Design project. He is solving for the stresses in a pin-jointed space frame by using relaxation techniques coupled with pattern recognition. Although Sketchpad III is not yet able to piece copies of an object (in this case, the individual bays of the truss) into large assemblies, only fifteen minutes were required to draw the repetitive truss. This time is at least an order of magnitude less than the time required to prepare punched cards describing the same truss. Clearly then, if computer program inputs can be described graphically, vast amounts of preparation time can be saved using Sketchpad III.

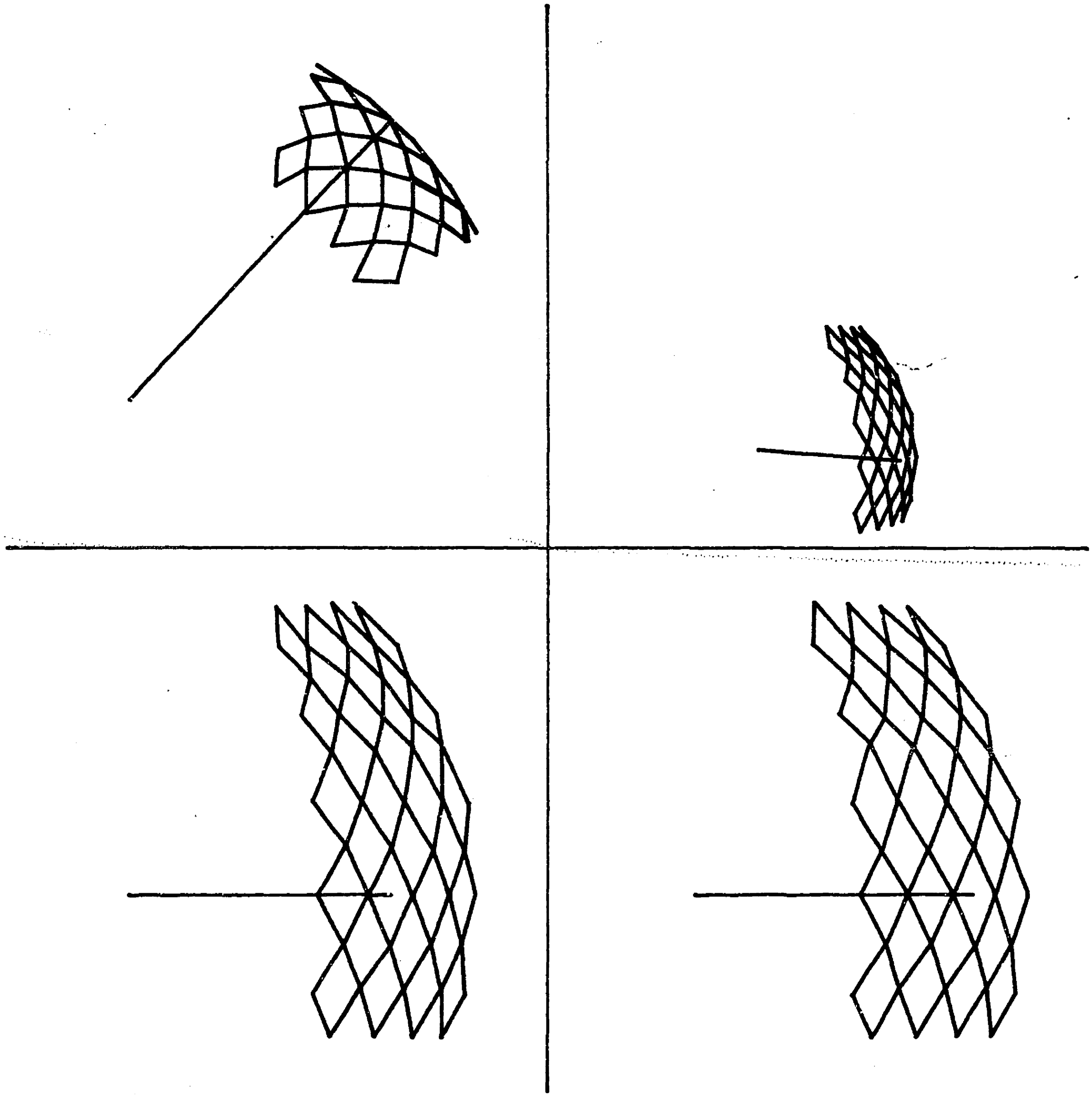


Fig. 14 Segment of a Radar Dome

To draw the segment of the radome shown in Fig. 14, a radius vector was established first so it appeared as a point in the side view. Once the depth of the PSL was placed at the end of the radius a single diamond was drawn in the side view: this established the plane of the diamond tangent to the radius vector. Rotating the radius vector through the angle subtending the arc of the diamond and maintaining the same depth assignment of the PSL, a second diamond was drawn abutting the first. In this fashion diamonds were drawn tangent to the sphere described by the locus of the radius vector. The perspective view is used exclusively for drawing cells that are small compared with the radius vector. Here, an enlarged perspective of the side view is selected enabling the micro drawing of a cell in the perspective while maintaining a macro view of the entire dome in the orthogonal views. The graphical description of the radome would become valuable when used as an input to other computer programs.

This last example points out one of the valuable uses of Sketchpad III--describing space frames that previously have been highly difficult to describe by drafting methods. This drawing took forty minutes: drawing a spherical frame on paper takes days. The ability to generate any auxiliary view of an object with the twist of a knob accounts for the vast savings in time.

B. CONCLUSIONS

1. Are Four Views Necessary?

Some observers questioned if all four views are necessary -- perhaps, they say, even one view would suffice if perspective and orthogonal views could be selected by pushbutton. After logging twenty hours of drawing experience with Sketchpad III, it can be safely said that one view is insufficient, two views are permissable, three views are desirable, and four views are useful.

One view drawing can be simulated with Sketchpad III, since any orthogonal view with or without perspective can be selected in the perspective viewing quadrant. Drawing in one view without simultaneous display of the object from another side does not allow the operator to check the depth at which he is drawing. Only by momentarily switching to another view by pushbutton can he determine the correctness of a depth assignment. The visualization difficulty is compounded when a new orientation suddenly appears on the screen: the viewer loses all sense of continuity offered by a moving rotation to the new position -- confusion is guaranteed. If gradual rotation is used for depth checking, the operator would be continually spinning a knob, since depth reinforcement is so vital.

Clearly, at least two views are necessary, a top and front view, for instance, with one of the views convertible to perspective. Depth checking and continuity problems would be eliminated. But two views are not enough. A third independent view must be added to give the necessary micro-macro viewpoint feature. Without this, detail work would be difficult as in the radome example where small cells are desired. A fourth view, as in Sketchpad III, offers the luxury of an additional viewpoint which aids visualization.

2. Do New Users have a Difficult Time Learning how to Draw with Sketchpad III?

For users who are previously acquainted with projection systems, learning how to use Sketchpad III is rapid. Usually ten to twenty minutes of instruction is necessary before an understanding, although not a confident skill, is gained. People who have had no experience with projection systems cannot use Sketchpad III without first learning projection conventions. However, even learning projection conventions will be easier with the aid of Sketchpad III.

Sketchpad III is capable of doing many things easily and well, but still falls short of general graphical communication. Research in the near future will bring the system closer to a general purpose system. Program additions that enable copies of structures to be locally manipulated will include the means of generating reflections for use in building symmetrical parts. Work is beginning on arbitrary surface manipulation, using Sketchpad III as a research tool.

The future will bring exciting changes in engineering. Two-way graphical communication networks between plants and field locations will enable engineers to change designs quickly as the need arises. Several consoles in a plant time-sharing one large central processor will permit several component designers to work concurrently on a system design. With bookkeeping programs, designers would be automatically informed if a design interfered with components described in other sections of the plant. Thus fuel lines would no longer be discovered passing through servomechanisms at the mock-up stage. With system designs becoming increasingly complicated, graphical communication with digital computers is bound to lessen the increasing amount of time spent in the design stage of manufacturing.

APPENDIX A

by

Alexander Vanderburgh

A BRIEF DESCRIPTION OF TX-2

At first glance, TX-2 is an ordinary single-address, binary digital computer with an unusually large memory. It is an experimental machine-many of its in-out devices are not commerically available. On closer inspection, one finds it has some important innovations - at least they were innovations at the time TX-2 was built (1956).

The distinctive features of TX-2 are:

1. Simultaneous use of in-out machines through interleaved programs.
2. Flexible, "configured" data processing.

Some other virtues include:

1. Automatic memory and arithmetic overlap.
2. A "bit" sensing instruction (i.e., the operand is one bit!).
3. Addressable arithmetic element registers.
4. Especially flexible in-out.
5. 64 index registers.
6. Indirect - i.e. deferred addressing.
7. Magnetic Tape Auxiliary Storage.

IN-OUT

The phase "simultaneous use of in-out machines" should be taken quite literally. It does <u>not</u> mean simultaneous control. Each unit has its own buffer register and only <u>one</u> of these can be processed by TX-2 at any given instant. It is the <u>relative</u> speed that is important. For example, the in-out instruction that "fills" the display scope buffer takes no more than 10 microseconds, but the display itself takes from 20 to 100 microseconds, i.e., up to ten times as long. While the display is busy, the computer can compute the next datum of course, but it can also initiate other in-out transfers. In practice, since most in-out units are much slower than their associated programs, the computer spends a significant percentage of the time just waiting (in "Limbo"), even when several devices are in use. Interleaved initiation of in-out data transfers is partly automatic and partly program controlled. Each in-out routine is independently coded and is operated by TX-2 according to its "priority." Each unit has a "Flag Flip-Flop" to indicate to control that it is ready for further attention. When a unit is ready for further attention its routine will be operated unless another unit of higher priority also needs attention. An index register is reserved for each in-out unit and is used as a "place-keeper" when its routine is not being operated. The sharing among in-out routines of storage, index memory, and the arithmetic element is the programmer's responsibility.

"CONFIGURED" DATA PROCESSING

The "normal" word length for TX-2 is 36 bits. For many applications 18 or 9 bits would suffice, and in some cases each piece of data requires the same processing. Configuration control permits "fracture" of the normal word into two 18-bit pieces, four 9-bit pieces, or one 27-bit and one 9-bit. These "subwords" are completely independent, for example, there are separate overflow indicators. In addition to "fracture" there is "activity" and "quarter permutation". Any quarter word can be made "inactive" i.e., inoperative. The 9-bit quarters of a datum from memory may be rearranged (permuted) before use. There are eight standard permutations, for example, the right half of memory can be used with the left half of the arithmetic element. Nine bits are required for complete configuration specification. Since only 5 bits are available for this specification in each instruction word, a special 32-word, 9-bit thin film memory is addressed by each instruction that processes data directly. A complete change to any of 32 configurations is therefore possible from instruction to instruction.

THE SMALLER VIRTUES

Overlap: TX-2 has two core memories - "S" memory, a vacuum tube driven 65,536 word core memory, and "T" memory, a transistor driven 4096 word core memory about 20% faster. Instruction readout can be done concurrently with the previous data readout if program and data are in separate memories.

The use of the arithmetic element is also overlapped. Instructions that follow a multiply or divide operation will be done during the arithmetic time if they make no reference to the arithmetic element. The overlap is entirely automatic and may be ignored if the programmer chooses. A careful programmer can gain speed by doing indexing after multiply or divide and by putting program and data in separate memories.

Bit Sensing Instruction: One instruction - SKM - uses a single bit of any memory word as its operand. Control bits provide 32 variations of skipping setting, clearing, and/or complementing the selected bit. This instruction can also cycle the whole word right one place if desired.

Addressable Arithmetic Element : Seventeen bits of the TX-2 instruction word are reserved for addressing an operand. This would allow a 131,072 word memory. TX-2 has only 69,632 registers of core storage. The toggle switch and plugboard memories, the real time clock register, the knob register (shaft encoder), and the arithmetic element registers use 55 of the remaining addressing capability. The arithmetic element registers are therefore part of the memory system and can be addressed, e.g., one can add the accumulator to itself.

Flexible In-Out: The TX-2 user must program each and every datum transfer. The lack of complex automatic in-out controls may seem to be a burden, but the simplicity of the system gives the programmer much more precise and variable control than automatic systems provide. For example, coordination of separate in-out units such as display and light pen is possible. Moreover, it is relatively easy to attach new in-out machines as they become available.

Index Memory and Indirect Addressing: Of the 64 index registers, one must devote a few to each in-out units program. With all 21 in-out devices concurrently in use, each program would have two index registers for normal programming use. In practice, one seldom uses more than half a dozen in-out units, and each routine would then have 9 - clearly a luxury. Indirect addressing provides a means for indexing normally nonindexable instructions, or for double indexing normal instructions.

Magnetic Tape Auxiliary Storage: Each TX-2 magnetic tape unit stores about 70 million bits, 34 times the capacity of the core memory system. Like a magnetic drum the tape is addressable. It can be read in either direction at any speed from 60 to 600 ips, and can be searched at a maximum of 1200 ips. It is used at present primarily for program storage. "Turn around time" - i.e. the time required to save one program and read-in a different one is seldom more than 2 minutes and often less than 30 seconds. (The read-in time, once the desired section of the tape is found, is about 12 seconds for 69,632 words.) A standard IBM 729 tape unit is also available.

SUMMARY OF VITAL STATISTICS - TX-2 DECEMBER 1962

Word Length:	36 bits, plus parity bit, plus debugging tag bit		
Memory:	256 x 256 core	65,536 words	6.0 μsec cycle time
	64 x 64 core	4,096 words	4.4 μsec cycle time
	Toggle switch	16 words	
	Plugboard	32 words	
Auxiliary Memory:	Magnetic Tape 2+ million words, 70+ million bits per unit (2 units in use, total of 10 planned)		
Tape Speeds:	selectable 60-300 inches/sec, search at 1000 inches/sec (i.e. about 1600 to 8000 36 bit words/sec)		

IN-OUT EQUIPMENT

Input:

Paper Tape Reader: 400-2000 6 bit lines/sec

2 keyboards - Lincoln writer 6 bit codes

Input:

Random number generator - average 57.6 μ sec per 9 bit number

IBM Magnetic Tape (Model 729 M6)

Miscellaneous pulse inputs - 9 channels - push buttons or other source

Analog input - Epsco Datrac - nominal 11 bit sample
-27 kilocycle max. rate

2 light pens - work with either scope or both on one

Special memory registers:

Real time clock

4 shaft encoder knobs, 9 bits each

592 toggle switches (16 registers)

37 push buttons - any or all can be pushed at once

Output:

Paper tape punch - 300 6 bit lines/sec

2 typewriters - 10 characters per second

IBM Magnetic Tape (729 M6)

Miscellaneous pulse/light/relay contacts - 9 channels (low rates)

Xerox printer - 1300 char. sec

2 display scopes - 7 x 7 inch usable area, 1024 x 1024 raster

Large board pen and ink plotter - 29" x 29" plotting area 15 in/sec slew speed. Off line paper tape control as well as direct computer control.

APPENDIX B

PUSHBUTTON CONTROLS

Button Name	Bit Number	Function
		--Construction--
Draw	1.8	Create a new straight line segment beginning at the position pointed at with the light pen. End of line remains attached to light pen.
Move	2.1	Graphical element (line or endpoint) moves with light pen.
Hold	4.9	Normal termination signal overridden. Motion merely suspended with pen flick. Used when moving pen to another view while drawing the same line.
Cylinder Mode	1.9	PSL to operate in cylinder mode; spherical mode operates when button is not depressed.
		--Deletion--
Delete	1.3	Graphical element pointed at removed from drawing.
*Delete Points	1.4	All unattached points in drawing are removed.
*Delete Drawing	1.5	Entire object is removed from memory.
Garbage	1.1	Compact list structure -- remove holes caused by above deletions.
		--Input-Output--
Punch	4.7	Punch paper tape of drawing for X-Y plotter.
Plot	4.8	Plot drawing on-line.

---

* Button for bit number 4.10 must be depressed simultaneously.

Button Name	Bit Number	Function
[*]IBM	4.3	Write contents of list structure on magnetic tape record. Number of record on tape given in toggle register 26. Typewriter confirms successful operation.
IBM	4.3	Read magnetic tape record. Same as above.
		--View Selections--
		Uppercase must be selected -- case shift button located next to left knob below scope. (Upper and lower case programmed to make room for 72 buttons, if ever needed.)
Orthogonal	4.4	Reduce perspective view to an orthogonal view. Turning the perspective knob restores perspective.
Standard	3.8	Perspective view to show standard orientation and magnification of object.
[**]Top	3.7	Perspective view to show top view. Perspective viewing distance set according to perspective knob position.
[**]Front	4.2	Perspective to show front view. Perspective viewing distance set according to knob position.
[**]Side	4.6	Perspective to show side view. Perspective viewing distance set according to knob position.
Reset Rotation	4.5	Reset rotation of object to original drawing position.
Translate	3.9	Drawing to translate with light-pen.

---

[*] Button for bit number 4.10 must be depressed simultaneously.

[**] If button for bit number 4.10 is depressed simultaneously, the resulting perspective view will correspond to a viewing distance inversely proportional to the scale of the object (corresponds to moving closer to the object as it is magnified, or moving away as it is reduced).

BIBLIOGRAPHY

1. Coons, Steven A., Notes on Graphical Input Methods, Memorandum 8436-M-17, Dynamic and Control Laboratory, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, Massachusetts, 4 May 1960.

2. Roberts, Lawrence G., Machine Perception of Three Dimensional Solids, Ph.D. Thesis, Massachusetts Institute of Technology, Department of Electrical Engineering, Cambridge, Massachusetts, June 1963.

3. Sutherland, Ivan E., Sketchpad, A Man-Machine Graphical Communication System, Ph.D. Thesis, Massachusetts Institute of Technology, Department of Electrical Engineering, Cambridge, Massachusetts, January 1963. (Also to be issued as an M.I.T. Lincoln Laboratory Report).

4. Smith, Albert F., Method for Computer Visualization, Technical Memorandum 8436-TM-2, Electronic Systems Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, September 1960.