

C H A P T E R O N E

HARDWARE FOR PRODUCING PICTURES

There are really only two kinds of pictures that can be produced by computer. By far the largest number of computer-produced pictures are line drawings: drawings which are mostly blank. A few pictures produced by computer, however, are half-tone images; that is, pictures like those produced on your TV set for which an intensity is defined at each of a quarter million or more points throughout the picture. We are going to talk here mostly about devices for producing line drawings. In some cases, these devices can be used also for producing half-tone drawings but generally at very low speed.

The two fundamental devices for producing line drawings by computer are the cathode ray tube display and the mechanical plotter. One should not overlook the fact, however, that ordinary line printers can be used to produce pictures. IBM, for instance, produces quite satisfactory circuit diagrams using standard symbols on a line printer. Knuth¹ used ordinary line printer symbols for producing block diagrams. If you have a line printer and no other display available, do not overlook its use for producing pictures.

Cathode Ray Tubes

Although cathode ray tubes were known prior to the Second World War they got their first major research and development efforts in connection with the wartime work on radar. The famous series of books published by the Radiation Laboratory at MIT reports some of the important understandings of

¹Knuth, Donald E., "Computer-Drawn Flowcharts", Communications of the ACM, Volume 6, Number 9, pp. 555-563, September, 1963

cathode ray tube technology that were developed there. During the war, the size of the tubes was greatly increased, the performance of phosphors was enhanced, and the construction of the electron guns was brought to a fine art. The P7 composite phosphor, for example, was developed specifically to make display of radar pictures possible.

The widespread use of television, of course, has put a cathode ray tube in nearly every home in the Country. Such mass production has greatly reduced the price of the kinds of cathode ray tubes used in television sets. Unfortunately, as we shall see, the low cost of television sets does not reflect itself in a corresponding low cost in computer display equipment because the requirements for a television picture tube and a computer display tube are quite different.

A cathode ray tube consists of five essential parts: (see Figure 1.)

1. a cathode structure which emits electrons. Because of the control grid, the electrons depart from essentially a point source.
2. an accelerating structure which causes these electrons to move rapidly down the tube,
3. a focusing structure which brings the beam of the electron into a more or less sharp focus on
4. the screen of phosphorescent material which makes the beam of electrons visible to the observer, and
5. an evacuated space for the electrons to move in.

It is not generally well understood that the beam of electrons in a cathode ray tube is not a narrow beam. In fact, the beam of electrons

diverges from the cathode to reach its widest point in the focusing structure. It is the purpose of the focusing structure to bring the divergent beam back into focus at the screen. The primary factor in controlling the size of the spot where the beam strokes the phosphorescent screen is the ratio of the distance between the cathode and the focusing structure and the distance between the focusing structure and the screen. Just as an optical lens magnifies or demagnifies according to the object and image distances, so the focusing structure in a cathode ray tube magnifies or demagnifies the size of the point source at the cathode if the cathode is closer to the focusing structure than the screen, as for example, in a short-necked large-screen TV tube, then the size of the spot on the screen will be relatively large. If, on the other hand, one wants to produce a tube with a very small spot, one should design it with a very long neck so that the cathode can be put far away from the focusing structure. In this way, the already small point source at the cathode will be demagnified to make an even smaller spot on the screen. It is possible to make cathode ray tubes with spots smaller than one-thousandth of an inch in diameter.

One of the measures of the quality of a cathode ray tube is the size of its spot. In general, cathode ray tubes with small spots are more expensive than those with large sloppy spots. It is not sufficient, however, to produce a tube which has a very small spot which can only be displayed in the center of the screen. The appropriate factor to consider for a CRT is the ratio of the spot size to the usable screen diameter. Thus, for example, a CRT with a 0.01 inch spot size and a 10 inch screen is equivalent in resolution to a CRT with a one-inch screen and a 0.001 inch spot size. The resolution of a CRT should be measured as the number

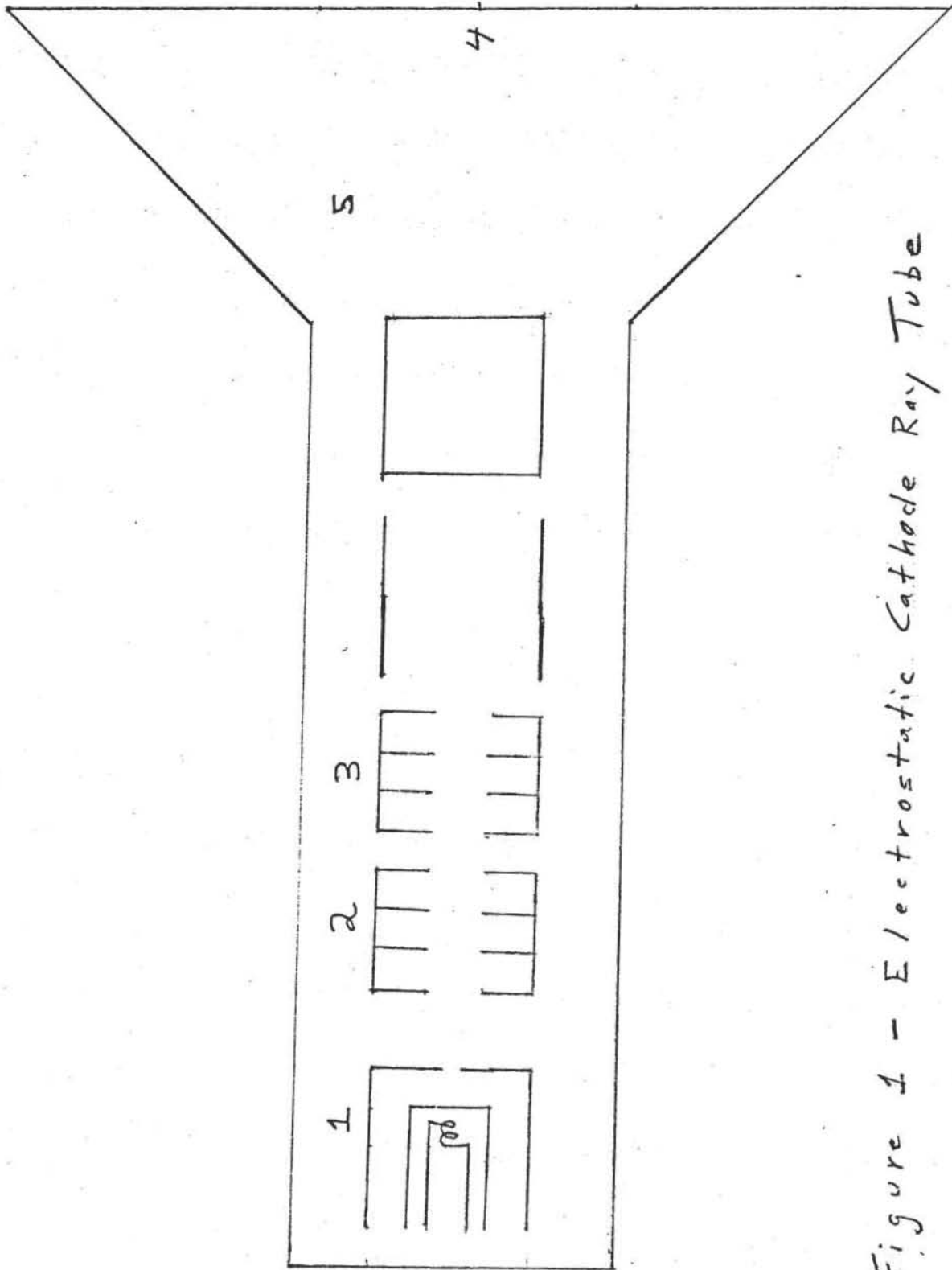


Figure 1 - Electrostatic Cathode Ray Tube

of lines that can be effectively displayed on the screen and NOT as a specific figure on the diameter on the spot. Cathode ray tubes which can display 500 lines are quite common in sizes from one inch diameter to two feet diameter. Cathode ray tubes which can display 5000 lines are barely obtainable, and then usually with a three inch or smaller screen.

Manufacturers of cathode ray tubes seem to have invented many ways to conceal the fact that the spots of their cathode ray tubes are larger than they would like them to be. Foremost among these is the use of the "shrinking raster method" for measuring spot size. The shrinking raster method of spot size determination is a relevant measure of the spot size in a tube for a television application. It is not, however, as a measure of spot size in a tube for a line drawing application. To measure spot size by the shrinking raster method, you display a raster of, let us say, 100 lines on the face of the cathode ray tube. You then decrease the gain of the deflection system so that the raster becomes smaller and smaller. At some point, the resulting display will no longer look like 100 lines, but rather like a uniformly lit rectangle. When the lines blend, you measure the size of the raster and divide it by the number of lines known to be in it. You announce this ratio as the spot size.

Now the spot of light produced by a cathode ray tube is not a clean, well-defined spot. The electrons leave the cathode with random velocities in random directions. Thus even were the focusing structure perfect, they would arrive at the screen in a randomly distributed block. Imperfections in focusing structure also shows up as random distributions in the intensity

of the arriving beam. Thus, if you plot the intensity of the light output as a function of position for the spot in a cathode ray tube, you will find some sort of bell-shaped distribution. (See Figure 2) The shrinking raster method measures the size of the spot based on two points very high up on this bell shape distribution.

In a line drawing display, however, the subjective width of the line will be based on cutoff points chosen subjectively much lower on the curve. In fact, according to data provided me by Sanders Associates for measurements on a particular tube, the shrinking raster measures line width at approximately the 75% intensity points whereas a subject view of the line width is taken at about the thirty percent point, a width which turns out to be nearly twice as great. (See Figure 3). So, whenever you hear or see a specification on a line width in a cathode ray tube, ask yourself, "How was this measured?".

How fast do the electrons in a cathode ray tube go? The velocity of an electron can be related to its energy by a simple equation.

$$E = \frac{1}{2}MV^2$$

but the kinetic energy of an electron obtained, is directly related to the voltage through which it has "fallen". One electron moving between two electrodes one volt different in potential receives one electron volt of energy.

Thus,

$$E = Qe$$

where Q is the electronic charge and e

Or:

$$V = \sqrt{\frac{20e}{M}} = K \sqrt{e}$$

is the potential

where K is approximately equal to 600,000 meters per second. (See Table I)

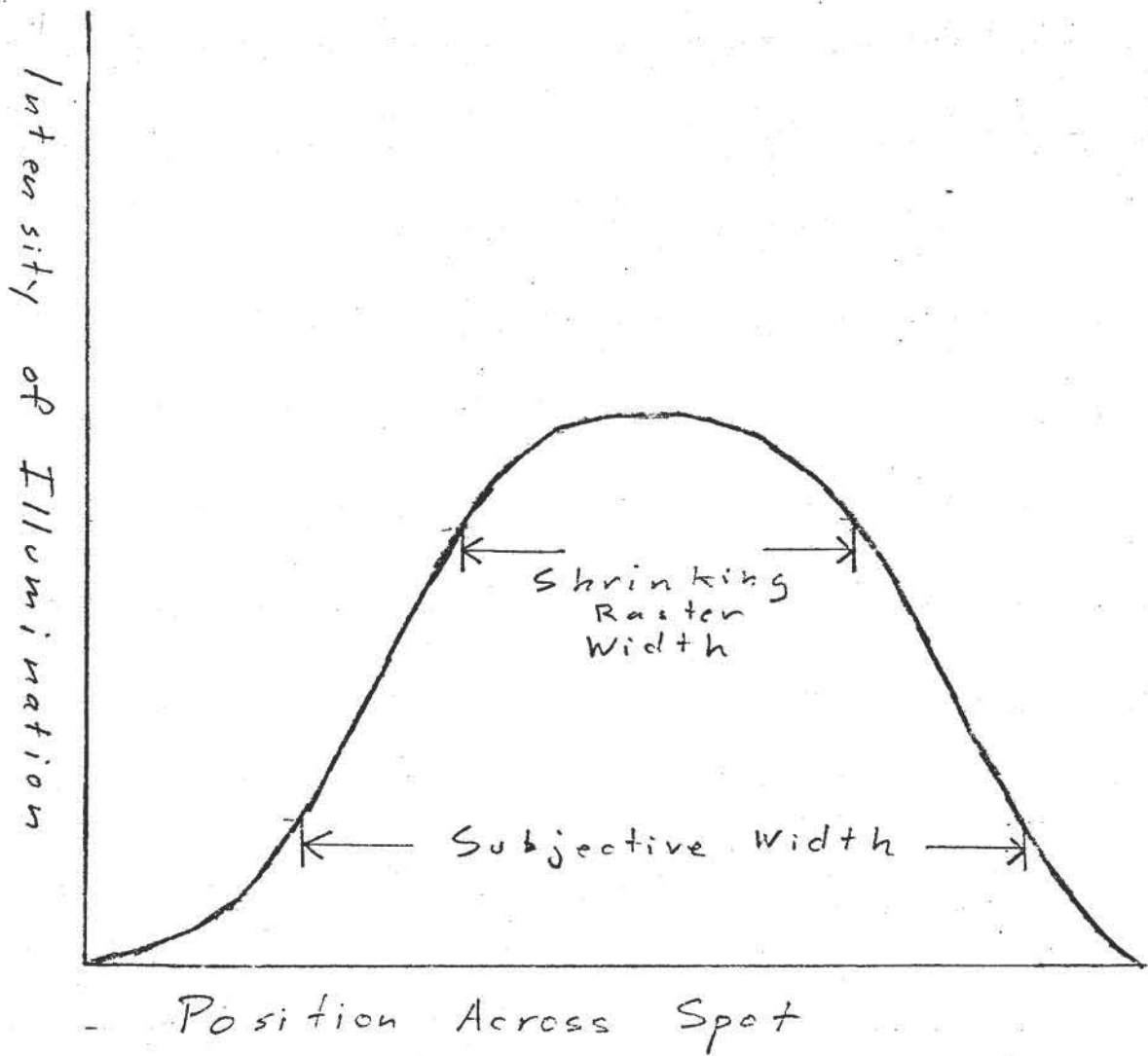


Figure 2 - Intensity Distribution -

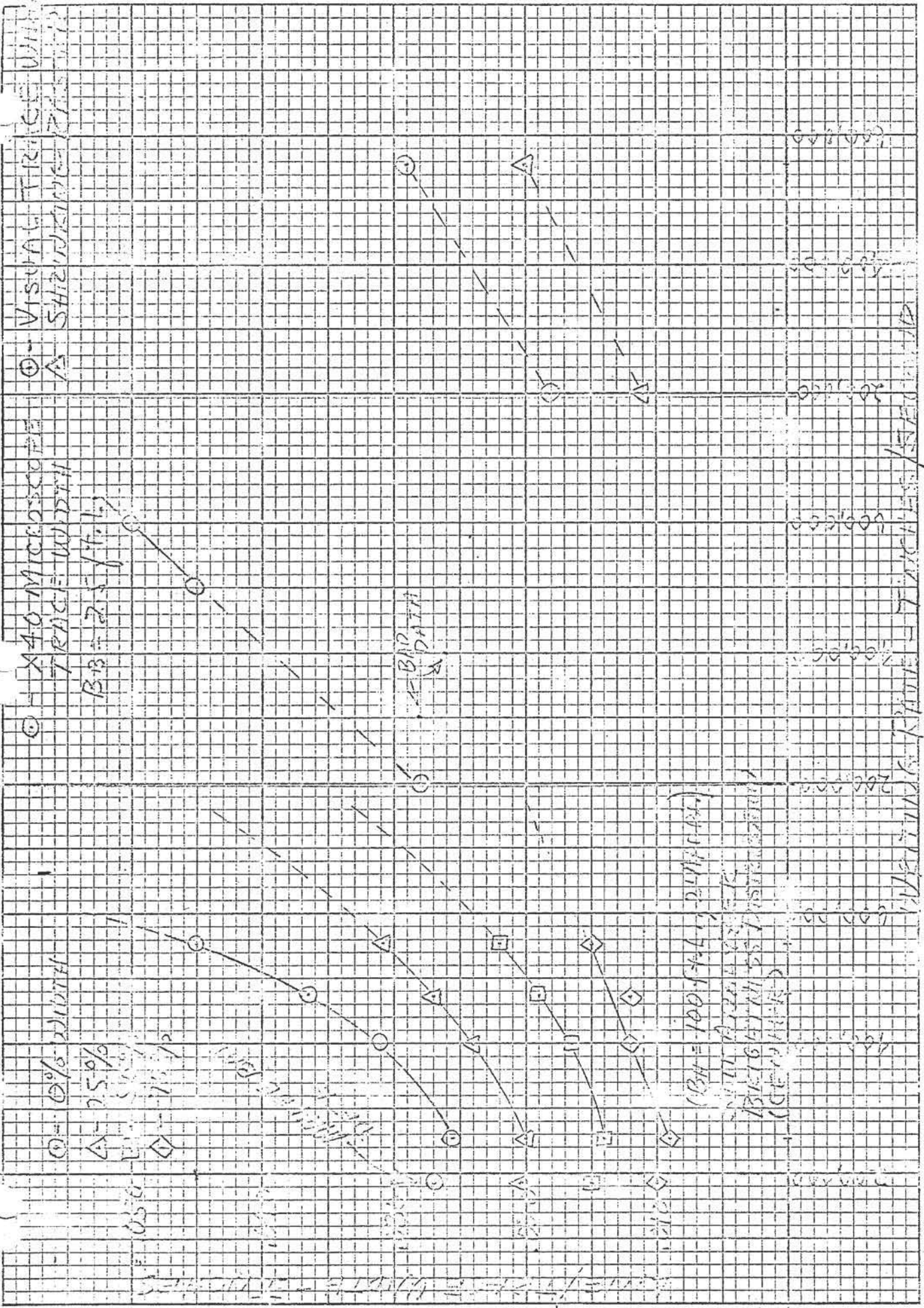


FIG. 1 RAULAND TYPE 6301L (MODIFIED), SER NO 4249 LIVES FOCUS



TABLE I
ELECTRON BEAM VELOCITY

<u>Accelerating Potential</u> <u>Volts</u>	<u>Speed m/sec</u>
1	$.6 \times 10^6$
100	$6. \times 10^6$
2,600	$30. \times 10^6 = c/10$
10,000	$60. \times 10^6$
118,000	$75. \times 10^6 = c/4$
500,000	$258. \times 10^6 = .863c$

Notice that although the higher energy electrons in the above table are fast, they are not yet relativistic.

Focusing and Deflection

The path of a moving electron may be modified by immersing it in an electric or a magnetic field. In order to focus the electron beam and in order to deflect it to different places on the CRT, we want to deflect the path of different electrons. Both electric and magnetic fields can be used for both focusing and deflection.

In order to focus the electron beam, we want to turn electrons far from the axis of the tube back toward the axis of the tube. In order to do this, we need a field which varies as a function of its distance from the axis of the tube, but not as a function of its angular position. To produce such a field electrostatically, focus electrodes of the form shown in Figure 4. may be used. To produce such a field magnetically, a coil is

wound around the neck of the tube such that its axis is parallel to the axis of the tube. Most commercial home TV sets use magnetic focusing. Many but not all cathode ray tube displays for computer use use electrostatic focusing.

If the face of the tube is flat, then the distance from the focusing structure to the screen will be different for spots in the center of the screen and spots at the edge of the screen. Thus a given setting of the focusing structure, while adequate to focus the beam in the center of the screen, may not be adequate to focus it at the edges. In addition to this, the deflection system itself may introduce errors in the focus of the beam. If such errors are objectionable, dynamic focus correction may be needed. In a display with dynamic focus correction, the voltage on the focus plates (if electrostatic focusing is used) or the current in the focusing coil (if magnetic focusing is used) may be changed as a function of the beam position. Because this correction is necessarily nonlinear (it is, in fact, approximately quadratic in beam position) dynamic focus correction is a nuisance. Moreover, in a cathode ray tube for computer display, dynamic focus correction must be done at very high speed. I know of no system for computer display which uses dynamic focus correction in a magnetic focusing system.

There are, of course, two ways to provide for deflection of the beam - magnetic and electrostatic. In a cathode ray tube with electrostatic deflection, four deflection plates are placed in two pairs just after the focusing structure. If a voltage is applied between the horizontal deflection plates, electrons passing between them will be

attracted towards the positive plate and repelled by the negative plate, and thus the beam will swing in the direction of the positive plate.

The angle of the deflection increases as more voltage is applied to the deflection plate, but if the electrons are moving faster, the effect of the deflection plates on them is less and so the angle of deflection is less. In mathematical equations,

$$\text{Tan } \alpha = \frac{Le_d}{2De_a}$$

e_a = accelerating voltage e_d = deflection voltage

L = length of deflection plates D = separation between
deflection plates

In a magnetic deflection system, deflection coils are used instead of deflection plates. The deflection coils are generally placed outside of the tube itself. Since the tube is made of glass and glass is non-magnetic, there is no need to go to the bother and expense of putting the coils inside of the evacuated envelope. The coils are generally found in pairs placed so that their axes are perpendicular to each other and the axis of the tube. The deflection of an electron by a magnetic field follows different equations from that in an electric field. In particular, the force deflecting the electron is a function not only of the magnitude of the field, but also of the velocity of the electron. Thus in a magnetic deflection system, if the electrons are moving faster, they will be pushed harder by the magnetic field. In mathematical terms,

$$\tan \alpha = \frac{LB}{\sqrt{\frac{2m}{Q} e_a}}$$

B = deflection field

e_a = accelerating voltage

L = effective length of
deflection field

m = mass of electron

Q = charge of electron

Contrast of Electrostatic and Magnetic Deflection

It is clear from the above equations that magnetic deflection is relatively more effective than electrostatic deflection for fast electron beams. Because of this, and because only moderate currents are required in the deflection coils whereas rather large voltages are required in the deflection plates, most home TV sets use magnetic deflection. On the other hand, the magnetic deflection coils store a great deal of energy when they carry enough current to deflect the beam very far. In order to swing the beam from one side of the screen to the other, this energy must be dissipated. Because of the large quantity of energy stored in the magnetic deflection system, it is inherently rather slow. Moreover, if magnetic materials are included in the deflection system to improve its performance as they usually are, then it is likely that removing the current from the coil will not remove all of the magnetism from the deflection system. Thus if the beam has been deflected to the right and we turn the current in the deflection coils off, the beam will stay slightly to the right of center. Whereas had it been deflected to the left and we turned the current off, it would stay slightly to the left of

center. This annoying non-return to desired position is called "hysteresis". Most magnetic-deflection computer displays suffer to a greater or lesser extent from hysteresis. ⁹ The construction of an electrostatic deflection system as parallel plates implies that a beam already deflected by horizontal plates will travel at an angle through the vertical deflection plates and thus be more influenced by them than a beam on axis. Thus in an electrostatic system, a beam deflected to the left tends to be deflected more up and down than a beam in the center of the screen. This distortion causes a square to be displayed with pointed corners and is known as pin-cushion distortion. Most electrostatic deflection systems suffer to a greater or lesser extent from pin-cushion. Another disadvantage of electrostatic systems is that the horizontal and vertical deflection points are not the same. A special kind of deflection structure called the deflectron (See Figure 5.) is used in some tubes to provide identical horizontal and vertical deflection centers. Finally, because a beam swinging toward or away from an electrostatic deflection plate feels fringe fields near the edges of the plate, electrostatic systems common^{LY} require dynamic focus correction. These desirable and undesirable properties of electrostatic and magnetic deflection systems are summarized in Table II.

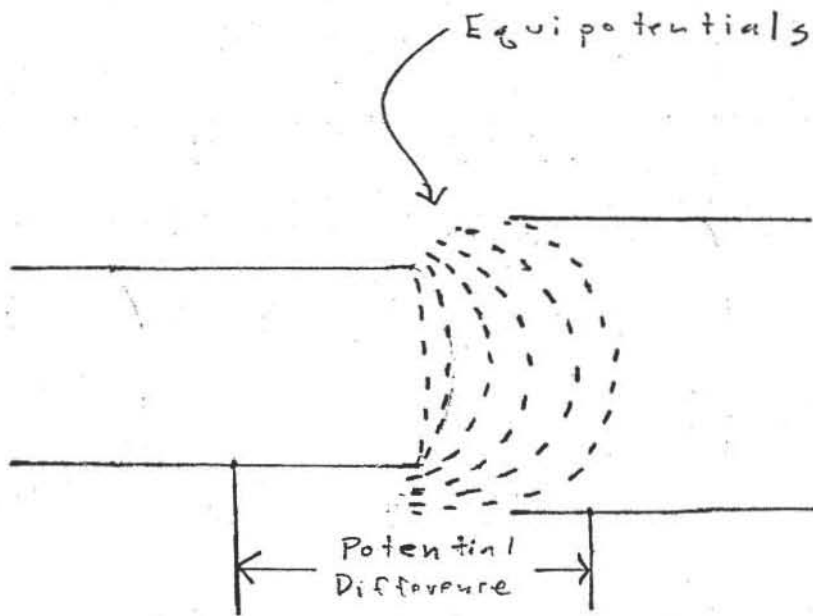


Figure 4 - Focussing Electrodes (Electron Lens)

See Appendix 3.

Schlesinger, Kurt, "Progress in the Development of
Post-Acceleration and Electrostatic Deflection",

Proceedings of the IRE, Volume 44, May 1956.

Figure 5 - Deflection

TABLE II

<u>M A G N E T I C</u>		<u>E L E C T R O S T A T I C</u>	
<u>Good</u>	<u>Bad</u>	<u>Good</u>	<u>Bad</u>
more efficient deflection for fast beams	high energy in field dictates slower speed, causes hysteresis	no hysteresis low power requirements	pin-cushion distortion defocussing in fringe field - needs dynamic focussing
deflection points same in x-y planes		high speed response	deflection points not the same in x-y planes
can be transistor-driven easily			require high-voltage devices for deflection plates

Phosphors

The electron beam or "cathode ray" in a CRT is invisible. The only reason for seeing a spot on the face of the screen is that a phosphorescent coating has been placed inside the tube. When electrons strike the phosphorescent coating, it glows. Different kinds of phosphors glow in different colors. In general, phosphors do not stop glowing immediately after the electron beam is turned off. Rather, they continue to glow for a longer or shorter time afterwards. Appropriate choice of phosphor material, then, can provide us with different colors and different lengths of afterglow in the spot. Because the phosphor material consists of individual grains of phosphor, usually one to a few thousandths of an inch in size each, cathode ray tubes with very small spot size must use carefully chosen phosphor if the inherent small size

of the electron beam is to be maintained in the visible spot.

The power in the electron beam arriving at the phosphor surface is the product of the voltage through which the beam has fallen and the beam current. Although the beam current is very small, (10 microamps is typical) the beam voltage may be very high. (10 kilovolts is common). Thus the beam of electrons arriving at the phosphor screen may carry a power of about 1/10 of a watt. Now 1/10 of a watt is not very much, but the spot is very small, and so the power density (beam power divided by spot size) may be quite high. For a 20 mill spot,

$$\frac{0.1W}{(20 \times 10^{-3} \text{ in})^2} = 250 \frac{W}{\text{in}^2}$$

The power density of 250 watts per square inch is considerably higher than the power density given off by an ordinary electric stove which, as you well recognize, runs red hot. Were the beams to stand still for any length of time, the phosphor coating on the tube might well be damaged. Computer displays commonly have burnt spots at the origin of their coordinate system.

The phosphor for a cathode ray tube gets into the tube as a liquid suspension. In fact, the tube envelope with the neck end not yet sealed looks, for all the world, like an Erlenmeyer flask. A half-inch or so of liquid containing the phosphor in suspension is sloped into this bottle. The tube stands for a day or so, and the phosphor settles out onto the inside of the screen. The relevant parameters of the phosphor are the grain size (which is usually only important in small-spot tubes),

the color of the glow, and the persistence. The light output of a phosphor decreases exponentially after the beam is turned off. The time constraint of this exponential decay may vary from a few microseconds to several seconds. What we would like, of course, is a phosphor which would continue to glow with uniform brightness for a fixed period of time and then suddenly extinguish. No known phosphor has this property.

The efficiency of a phosphor may vary over a wide range. Two kinds of efficiency need to be considered. First, the number of photons the phosphor will emit per incoming electron, and secondly, the visibility of the resulting light. A phosphor which emits infrared energy is not much use in visual display. The phosphors wear out with continued use. This wear shows up as decreased efficiency as the phosphor ages. High performance phosphors, that is, phosphors with very high efficiency and consequently high light output are particularly prone to aging.

Phosphors can be effectively mixed to get useful effects. For example, the P7 phosphor developed during the war is, in fact, a double-layer phosphor. The layer nearest the observer glows yellow when bombarded with ultraviolet light. The inner layer glows ultraviolet when bombarded with electrons. The combination of two layers proved to have greater efficiency than any single phosphor we could have found.

Standard phosphor types are given number designations, such as P7, P11, P40, etc. In order to be assigned a number, a phosphor must be adequately described by its manufacturer. Descriptions of the persistence, efficiency, and color of various standard phosphors are available.

(See reference 4.)

References

1. Spangenberg, Karl R., Vacuum Tubes, McGraw-Hill Book Company, Inc., (New York, 1948), pp. 104-107 and Chapter 15.
2. Advances In Electronics, Volume II, Academic Press, (New York, 1950), Chapter 1.
3. Schlesinger, Kurt, "Progress in the Development of Post-Acceleration and Electrostatic Deflection", Proceedings of the Ire, Volume 44, May 1956.
4. "Phosphor and Persistence", Computer Display Review Revised to July 1967, pp. II.13-II.18.

C H A P T E R T W O

ARITHMETIC AND GRAPHICS

Fixed Point Computations.

Many of the computers with which computer graphic equipment is used can do only fixed point arithmetic. Even on machines with floating point arithmetic units, many of the computations for computer graphics' problems are done in fixed point arithmetic. For this reason, we will first review fixed point arithmetic operations. In the following parts of this book we will often assume that arithmetic is being done in fixed-point. In this chapter we will first see how numerical quantities are represented for fixed-point fractional arithmetic. We will then see how addition can replace multiplication in many simple operations such as drawing lines and curves.

In a fixed-point binary computer, one is free to choose any position for the binary point provided that the chosen position is used consistently. One commonly chosen position for the binary point is at the right hand end of the word so that all numbers are thought of as integers in the range $-2^N \leq X \leq 2^N$. The left equality applies only in two's complement machines. Another commonly chosen position for the binary point is immediately to the right of the sign bit which is the lefthand end of the word. Numbers all are thought of as fractions in the range $-1 < x < 1$. In two's complement

machines it may be possible to represent -1 exactly.

If the integer form of representation is used, then the binary positions assume their familiar values:

SIGN	.	.	.	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
------	---	---	---	-----------	-----------	-----------	-----------

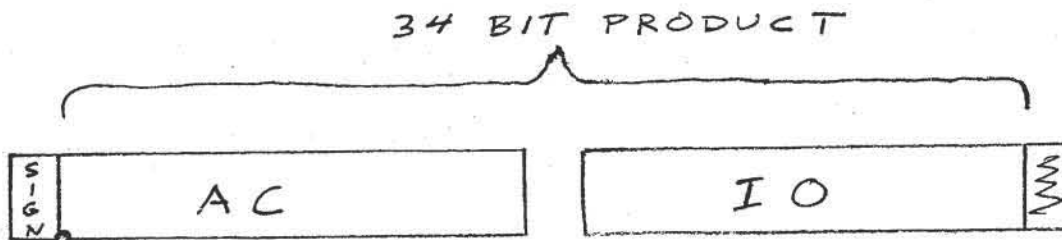
If, on the other hand, the fractional representation is used, then the positions of the binary fraction assume the values

SIGN	$2^{-1} = 1/2$	$2^{-2} = 1/4$	$2^{-3} = 1/8$	$2^{-4} = 1/16$.	.	.
------	----------------	----------------	----------------	-----------------	---	---	---

Thus, for example, the binary fraction 0.101 represents $1/2 + 1/8$ or $5/8$.

Whether numbers are thought of as integers or fractions can often make a difference in the case of understanding and using arithmetic instructions. For addition and subtraction, it makes not the slightest difference which representation is used so long as the chosen representation is used consistently. If the sum of two numbers exceeds the largest representable number, 2^N or 1, as the case may be, overflow will result. On the other hand, many fixed-point machines such as the PDP-1, SDS-940, and TX-2 have multiply and divide instructions for fractional arithmetic. The product of two 18 bit numbers, (that is, 17 bits and a sign bit) is a 35 bit number (that is, 34 bits and a sign bit). In the PDP-1 multiply instruction the product occupies the sign and seventeen positions of the accumulator and seventeen positions of the IO register. The least significant position of the IO register is not used for product information.

Multiplication leaves the most significant part of the product in the accumulator and the least significant part left justified, in the IO register. If you think of numbers as binary fractions, no shifting need be done to make use of the most significant part of a product. If, on the other hand, you think of numbers as integers, then the product appears to be shifted one bit to the left in the IO register. In order to use the product as an integer, you must shift it to the right. It is plain, then, that the multiplication and division instructions of such a machine are most easily used if numbers are thought of as fractions.



Because fixed-point multiplication is usually fractional in nature, I suggest that you think of numbers as fractions wherever possible. To put numbers in fractional form, use "normalizing factors" in setting up the problem. If, for example, you wish to represent a distance which may have a maximum value of 3 feet, represent all distances in yards. If you wish to represent a distance with a maximum value of 300 feet, divide all distances by 300 to obtain a normalized distance which varies between 0 and 1. You may think of the normalization procedure as picking appropriate units in which to represent quantities. Thus, in our 300 feet example, we can assume

some peculiar distance measure, (say the "George" = 300 feet) and represent distances in "Georges". You can also think of numbers in the computer as representing the fraction of the maximum deviation possible. On a computer display, for instance, it is convenient to represent the left-hand edge of the scope as -1 and the right hand edge as +1. Thus the number "1/2" when plotted on the scope means "half the distance from the center to the right-hand edge", i.e., one-half the maximum deviation to the right.

If numbers can be represented only in the range $-1 \leq X \leq +1$, then the sum of two numbers may possibly be out of the range. Fixed point computers will detect this condition with overflow. A useful trick to use to avoid overflow problems is that the average of two numbers will always be within the range. Thus if you follow an addition by dividing by two, you can avoid the overflow problem entirely.

In most fixed-point machines, the format for the division instruction (which requires a double length numerator) is identical with the format that is the result of the multiplication instruction. Thus it is convenient to precede each division with a multiplication. One can, for instance, divide by a normalizing factor after each multiplication. Suppose, for example, that we have some number x which we wish to square. If x is a relatively small number, say it has six leading zeros, then x^2 will have twelve leading zeros and can only be represented in an eighteen bit machine to six bit precision. On the other hand, if we multiply by x and then divide by a normalizing factor so that we represent not x^2 but $\frac{x^2}{n}$, where n is a typical value of x , the resulting quotient will have only about the same

number of leading zeros as has x itself. Between the multiply and the divide, the temporary product will be represented with 34 bit precision.

Digital Differential Analyzers

Suppose we want to draw 100 points P_i on a line from a point "R" (x,y) to a point "S" (x,y) . The points to use are $R, R + \frac{S-R}{100}, R + 2\frac{(S-R)}{100}, \dots, S$. In other words, we can move from R to S in small increments, each of which is some fraction of the vector $S-R$. Repeated addition of a single small vector will generate successive points on the line.

$$P_1 = R$$

$$P_i = P_{i-1} + \Delta P$$

(1)

$$\text{where } \Delta P = \frac{1}{N}(S-R)$$

The representation of the small vector, however, will have to be very precise. If any bits are lost in forming $\Delta P = \frac{1}{N}(S-R)$, they will show up as an error in the position in which the line arrives at S. Equipment to implement equation (1) might be:

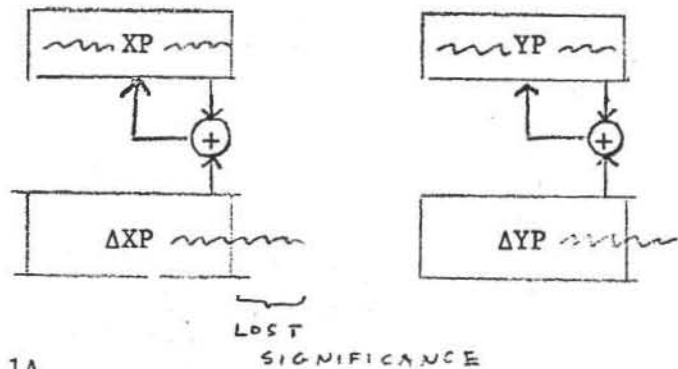


Figure 1A

In order to avoid cumulative errors which might be caused by truncating ^{the} small vector $\Delta = \frac{1}{N}(S-R)$, we might extend all our registers to the right as shown in figure 1b.

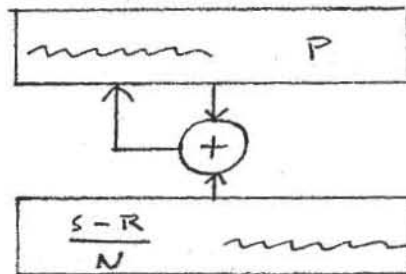


Fig. 1 b

The addition may then be performed with full accuracy. Unfortunately, however, we are now using many more bits in our registers than we actually need because we know that Δ is going to have many leading zeros, and we need not actually represent them in hardware. Thus, the representation shown in Figure 1c would be adequate. Successive additions will still be done with the full precision required.

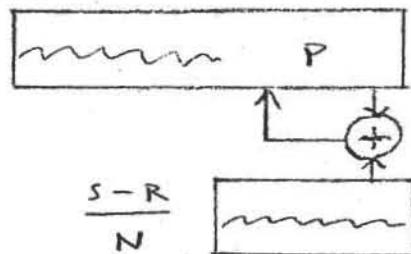


Fig. 1 c

The upper register in Figure 1c may be thought of as being separated into two parts, a most significant part, suitable for representing the position on the display, and a least significant part suitable for holding increments as shown below.

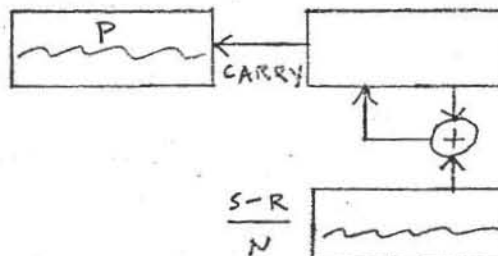


Fig. 1 d

A carry path is implemented between the two parts of the register, as it was in Figure 1c, so that overflow from the least significant part of the addition will increment the more significant part of the upper register. Nothing has been changed between figure 1c and Figure 1d except that in Figure 1d the connection between two flip-flops at the center of the long register has been shown explicitly. We can, of course, separate the two halves of the long upper register physically or conceptually provided that we continue to provide the carry path between them.

If we think of the two halves of the long register as entirely separate, we get a new view of what the incremental line-drawing operation is doing. It is customary to call the least significant half of the long register the "remainder" register "R" as shown in Figure 1E. Initially, the remainder register contains zero. During each operation of the device, the content of the Δ register and the remainder register are added together and left in the remainder register. If an overflow is generated from this addition, "1" is added to the P register. This operation is, in fact, identical to the additions implied in Figures 1a and 1b but provides the basis for a conceptual separation between the adder portion (registers R and Δ) and the counting portion (the register P).

To see how such a device works with actual numbers, suppose that the Δ register contains a number which is one-half of the maximum representable

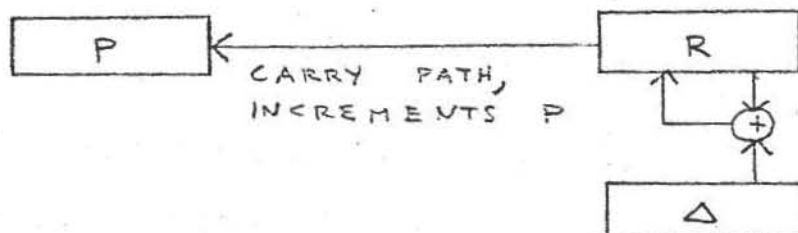


Fig. 1e

value. Then the first addition of Δ to R will leave the number $1/2$ in R. The second addition of Δ to R will cause an overflow and leave the number zero in R. The third addition will leave the number $1/2$ in R, the fourth addition, zero, and so forth. Alternate additions will cause P to count. In other words, with $1/2$ in the Δ register, the register P will count at $1/2$ the maximum rate. On the other hand, if the largest possible representable number is put in Δ , then overflows in addition will occur on every cycle, and P will count at its maximum possible rate. In other words, P is counted at a rate exactly proportional to the fraction represented in the register Δ . Moreover, because the R register can be thought of as the least significant portion, or a right-hand extension of P, the successive values in the P register will be uniformly distributed.

From a grosser point of view, then, a digital differential analyzer is a device which accepts a stream of add commands and converts them into a stream of carry pulses at a lower rate. The rate of output pulses is determined by the magnitude of the numbers stored in the device. In other words, the digital differential analyzer multiplies a stream of pulses by a fraction less than one to produce another stream of pulses, fewer in number but uniformly spaced in time.

The Binary Rate Multiplier

The intent of a binary rate multiplier (BRM) is identical with that of the differential digital analyzer (DDA) described above, namely to provide a series

of output pulses fewer in number than the input pulses. Unfortunately, the binary rate multiplier, though simpler in design, produces a sequence of output pulses which are non-uniformly spaced. Nevertheless, for some applications, it is a useful device.

A binary rate multiplier consists of two registers, a counter register (C) and a mask register (M). The bits of the counter register and the bits of the mask register are identified with the least significant bit of the mask register, and the most significant bit of the mask register is identified with the least significant bit of the counter register. Whenever a particular bit of the counter register changes from 0 to 1 and the corresponding bit of the mask register is a "1", an output pulse is generated. Logic to implement such a device is very simple.

Operation of the binary rate multiplier depends on an important property of binary counting: namely that only one bit of a binary counter ever changes from zero to one. Examination of the binary sequence shown on page 9A will quickly verify this property.

<u>COUNT</u>	<u>OUTPUT PULSES</u> <u>FOR 3/4</u> <u>(M) = 0011</u>	<u>OUTPUT PULSES</u> <u>FOR 5/8</u> <u>(M) = 0101</u>
0 0 0 0 X	X	X
0 0 0 1 X	X	X
0 0 1 0 X	X	X
0 0 1 1 X		X
0 1 0 0 X	X	X
0 1 0 1 X	X	X
0 1 1 0 X	X	X
0 1 1 1 X		
1 0 0 0 X	X	X
1 0 0 1 X	X	
1 0 1 0 X	X	X
1 0 1 1 X		X
1 1 0 0 X	X	X
1 1 0 1 X	X	
1 1 1 0 X	X	X
<u>1 1 1 1</u>		
16 Counts	<hr/> 12 Counts	<hr/> 10 Counts

Moreover, the least significant bit of a binary counter changes from zero to one every other count as marked by an "X" in the figure. The next most significant bit changes from zero to one every fourth count, the next most significant every eighth count, and so on. Thus, if the most significant bit of the mask register contains a one representing the binary fraction $1/2$, output pulses will be generated for exactly half of the steps. If both the two most significant bits of the mask register are one, representing the binary fraction $3/4$, output pulses will be generated during three counts out of four. One can easily verify that the number of output counts generated during one complete cycle of counting will be exactly the number represented in the mask register.

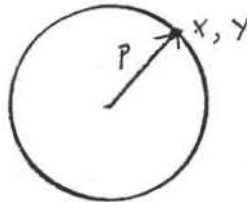
The binary rate multiplier produces pulses which are not in uniform time sequence. One might well ask, then, what the maximum cumulative error of the output pulse count is. Suppose, for example, that we accumulate the output pulses in a counter. What is the maximum discrepancy between the content of that counter and the correct content of the counter driven by, for example, a DDA. The output stream of pulses from a binary rate multiplier is incorrect by at most one pulse. Such accuracy might lead us to believe that a binary rate multiplier used for generation of lines in a display would provide adequate accuracy. Unfortunately, the maximum discrepancy between two binary rate multipliers with different contents is larger, and because the discrepancies occur in irregular ways, lines drawn by binary rate multiplier line-drawing devices appear to be unpleasantly irregular. The DEC Type 340 and 338 displays use binary rate multipliers for line generation. The worst case lines are drawn

for Δx and Δy values approximately complementary, e.g. 254_8 and 422_8 .

If binary rate multipliers are connected together in sequences such that the output of one drives another, the resulting output pulses may be very non-uniform. Binary rate multipliers are therefore useful only for very simple operations.

Drawing Circles Incrementally

Suppose that we wish to generate a series of points on a circle. Suppose, with no loss of generality, that the circle is to be centered at the origin of our coordinate system. The initial point on the circle $P(x,y)$ is positioned as shown in the diagram.



Obviously if P is considered as a vector, it represents a radius Vector of the circle.

If we wish a point adjacent to P on the circle, we should move a small distance from P approximately at right angles to the radius vector. Analytic Geometry tells us that to move at right angles to a given vector we interchange its x and y components and change the sign of one of them. Therefore, the vector $s = \epsilon y, -\epsilon x$ will be a small vector at right angles to P . This reasoning suggests that simple difference equation

$$X_{i+1} = X_i + \epsilon Y_i$$

$$Y_{i+1} = Y_i - \epsilon X_i$$

(1)

will generate successive points on the circle.

Unfortunately, successive points generated by equation (1) each lie slightly further from the circle than their predecessors, as analysis of the geometry will show. The radius vector for each new point is the hypoteneus of a little right triangle one side of which was the radius vector for the preceding point. Circles drawn using the difference equation (1) grow approximately π times the size of the unit step in radius per revolution.

If equation one is represented in matrix terms, it appears almost to be a rotation of coordinates.

$$\begin{bmatrix} X_{i+1} \\ Y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & \epsilon \\ -\epsilon & 1 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \end{bmatrix} \quad (2)$$

The determinant of the rotation matrix, however, is slightly greater than one, which means that successive applications of equation one increase the scale of vectors so transformed.

A small change to equation one will produce a different equation capable of drawing perfect circles. Instead of applying the x and y portions of the difference equation simultaneously, they are applied successively. Applying the separate equations successively is, of course, just what one wants to do on a digital computer, because less memory is required. The improved form of the equations is:

$$X_{i+1} = X_i + \epsilon Y_i$$

$$Y_{i+1} = Y_i - \epsilon X_{i+1}$$

(3)

which can be converted to the matrix form as was done above. In the matrix formulation:

$$\begin{bmatrix} X_{i+1} \\ Y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & \epsilon \\ -\epsilon & 1-\epsilon^2 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \end{bmatrix}$$

(4)

The determinant of the matrix in equation (4) can be seen to be unity which implies that no scale change is involved. In fact, computer implementations of equation (3) are known to produce circles which close.

Notice that the equations for generating the circle given above require only shifting and addition. In fact, the PDP-1 computer program

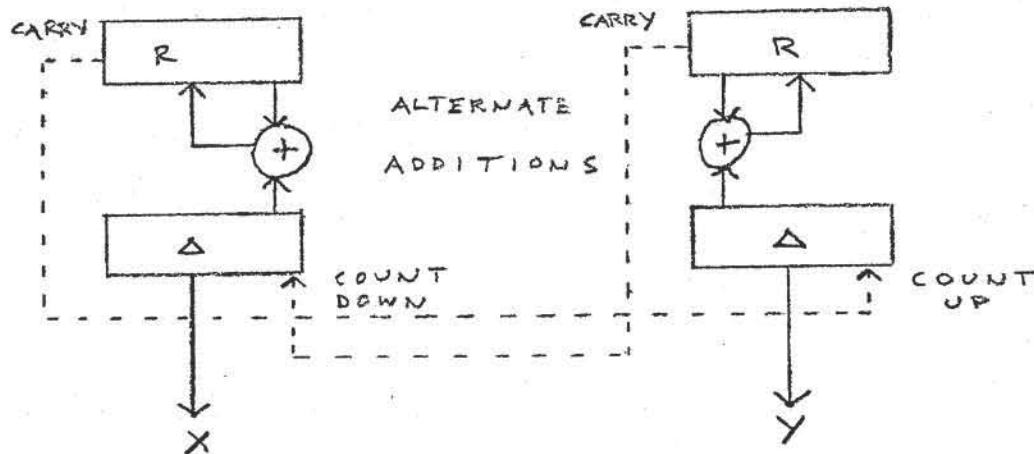
```

start,    dzm count
          lac x
          lio y
          dpy
          sar step
          cma
          add y
          dac y
          sar step
          add x
          idx count
          sas arclength
          jmp start + 1
          .
          .
          .

```


will generate successive points on the circle.

Equation number three can also be implemented in digital differential analyzer terms. The equipment shown below will generate successive points on a circle.



Anyone familiar with analog computers will at once recognize this configuration as similar to the two-integrator set-up one would use to generate sines and cosines on an analog computer. In fact, the DDA add element is similar in many respects to the integrator commonly used in analog computers. Large systems for function computations and complicated navigation equipment are frequently built of DDA elements.

References (On Reserve Book List)

1. PDP-1 Handbook (DEC) - Add, Multiply, Divide instructions
2. Forbes, George F., Digital Differential Analysis
3. Haring, Donald, Analysis, and Simulation of Incremental Computations Performed by Binary Rate Multipliers
4. Stotz, Robert H., Specialized Computer Equipment...Curvilinear Figures
5. Sutherland, Ivan E., Sketchpad: A Man-Machine Graphical Communication System

CHAPTER THREE

"WINDOWING"

It is often convenient to treat a drawing in a computer as much larger than the face of the scope. The scope face, being only about 10 inches square, is too small to represent a complex drawing. Most engineering drawings are made on paper 17" by 22" (size C) or even 22" by 34" (size D). Moreover, the resolution available in a computer word, even a word as short as 18 bits, is far finer than the resolution available on the scope face. It is therefore possible to represent far more information digitally in the computer than can be displayed at a single time on the face of the scope.

In this section we will consider the computations required to present a portion of a stored drawing on the face of the scope. More important, we will consider the computations required to eliminate from view the parts of the drawing not visible. Because the difficult part of the job is to cut out the portions of the picture not being seen, the task is sometimes called "Scissoring". Because information can be selected for display not only on the basis of the geometry, we will discuss here, but also on the basis of meaning, I prefer to call this task "Windowing". I believe that windowing is fundamental to good use of a cathode ray tube display. I believe that in virtually all display programs, the scope should be able to display information selected from the total information available in memory. The scope should be thought of as a window through which one can examine the material in the computer.

Positional information stored in the computer must be related to some coordinate system. Let us call that coordinate system the "page" coordinate system. Let us think of page coordinates as running from -1 to +1 in each axis; in other words, we will think of coordinates in the page coordinate system as signed fractions of the maximum representable coordinate, whatever that may be. Let us refrain from assigning a particular size to the page coordinate system because the dimensions represented might be astronomical units if we are looking at pictures of star maps, or microscopic units if we are looking at mechanical drawings of integrated circuits or dollars versus time for a cost accounting chart. For convenience sake, I think of the page coordinate system as being quite large, say the size of a wall, but let me emphasize again that it may actually have any kind of dimensions at all, depending on the problem at hand.

The scope has a coordinate system of its own which can also be thought of as running from -1 to +1 in each axis. Numbers in the scope coordinate system are thought of as fractions of the maximum useful number, i.e. as fractions of the number that represents the edges of the screen. Scope hardware is usually capable of accepting numbers of only ten or eleven bits for each axis. The ten or eleven bits are generally positioned at one end or the other of the computer number format. If the bits selected are at the left of the computer format, the scope numbers are easily thought of as signed fractions; dropping the unused right-hand bits merely decreases the resolution but positions outside the scope area cannot conveniently be represented. If the bits selected are at the right of the computer format, the scope numbers are

easily thought of as integers between 0 and 2^n-1 , say 1023; dropping the unused left-hand bits remaps spaces outside the scope area onto the scope face. Neither scope coordinate system is better than the other. Conceptually, however, it is useful to think of scope coordinates as signed fractions of the maximum representable coordinate regardless of what actual format is used. The windowing job, then, is to take a portion of the page coordinate system and display it on the scope as shown in Figure 1. If the portion chosen includes the entire page coordinate system, then we will look at the entire drawing. If the portion to be displayed is but a small fraction of the page coordinate system, then that small section of the picture will be spread out to cover the entire scope, and we will look at a magnified view of the drawing. By controlling the position and size of the portion of the page to be observed, we can control what part and how much of the picture we observe on the screen. The window can be thought of as a fictitious box appearing in the page coordinate system. Everything inside the window is to be shown on the screen; things outside the window are not to appear. The position of the window is described by WC_x and WC_y , two numbers which are represented in the PAGE COORDINATE SYSTEM. In the illustration, WC_x is about $5/8$, and WC_y is about $1/8$ (of the maximum representable page coordinate number). Because the window need not be square, its size is described by two numbers, WS_x and WS_y , also numbers represented in PAGE COORDINATES. In the illustration, WS_x and WS_y are both equal to about $1/4$ (of the maximum representable page coordinate number).

The transformation implied in Figure 1 is very simple. To find the scope coordinates X_s and Y_s of a point $P (X_p, Y_p)$ on the page, we have merely to find where that point rests with respect to the window.

In other words, is that point to the right or left of the center of the window and what fraction of the maximum window size is its separation from the center of the window. In other words,

$$X_s = \frac{X_p - WC_x}{WS_x} \quad Y_s = \frac{Y_p - WC_y}{WS_y} \quad (1)$$

Notice that the units of all symbols on the right of equation (1) are the same. The numerator is the difference of two numbers represented in page coordinates, and thus is in page coordinates, as is the denominator. The result of the division is therefore unitless. It represents the signed fraction of full-scale deflection within the window.

It is often useful to display picture material on a smaller portion of the scope than the full screen. I will call such a portion of the scope a "viewport". There might be several viewports on the scope each one displaying a different set of information from the picture or from separate pictures. For instance, one viewport might contain an overall view, and another viewport an enlarged section as shown in Figure 2. The position and size of the viewport can be described in the same way as the position and size of a window, but in scope coordinates because the viewport is to be positioned on the scope.

The transformation now required to put material from the page through a window into a viewport on the scope is:

$$X_s = \frac{X - WC_x}{WS_x} VS_x + VC_x \quad Y_s = \frac{Y - WC_y}{WS_y} VS_y + VC_y \quad (2)$$

The dimensionless number from equation one (which represents the signed fraction of deflection within the window) has been multiplied by the

size of the viewport (to give the signed deflection within the viewport) and then offset by the center position of the viewport. Notice that equations (2) contain a multiplication and a division. As was pointed out in the chapter on fixed-point computations, most fixed-point computers are so arranged that divisions can conveniently follow multiplications. An appropriate way to implement equations (2) then, is to do the multiplication before the division.

Because the format of information for transfer to a display scope is often different from the internal representation of numbers in the computer, it is often convenient to think of the windowing transformation in terms of a viewport even if the viewport used is the entire scope. A full-scope viewport can be thought of as that portion of a large (-1 to +1) fictitious scope which is actually occupied by the real scope. Thus equations (2) are useful for transformation of information even if no visible viewport is intended.

The windowing operation is essentially non-linear. Material which is outside the window must be eliminated from view and not merely transformed as indicated in equations (2). In terms of equations (1), material must be eliminated if the division results in a number larger than one. In most computers, a fractional division which results in a larger number than one generates an overflow indication. Such an overflow is a useful means of discovering which information must be rejected. If the picture consists only of points, it is sufficient to reject any point for which the division of equation (1) results in overflow. If such points are not rejected, but merely truncated so as to fit into the word format for the display, the points will appear to be wrapped around toroidally on the scope.

Incidentally, a program which merely masks off bits to the left of those actually used by the scope or which ignores overflow treats the scope as a toroidal space. If such a program plots a succession of points with increasing y coordinates, the point after the point at the top of the screen will be at the bottom of the screen. For such a program, points at the top and bottom of the screen are adjacent, as are points at the left and right edges. Because the top and bottom edges of the screen are adjacent (that makes a cylinder) and the right and left edges of the screen are adjacent (twisting the cylinder into a doughnut) the space is described as toroidal.

Although windowing for points is accomplished simply by detecting overflow in a division, windowing for lines is not quite so easy. A straight-line segment can, of course, be represented by the coordinates of its two endpoints. If both ends of the line are contained within the window, (Figure 3A) then the entire line will appear on the screen. If one end of the line is in the window, then only a part of the line must be displayed. In such a case, one must merely compute where the line leaves the window.

If, on the other hand, both ends of the line are outside the window, it may be possible to reject the line entirely, as shown in Figure 3B. If both ends of the line are to the right, or to the left, or above, or below the window, then the line can be rejected entirely. If however, the ends of the line lie off the screen in different directions as shown in Figure 3C, then the lines may or may not pass through the screen.

Some geometry is required to determine whether or not such lines show at all.

WINDOWS AND INSTANCES

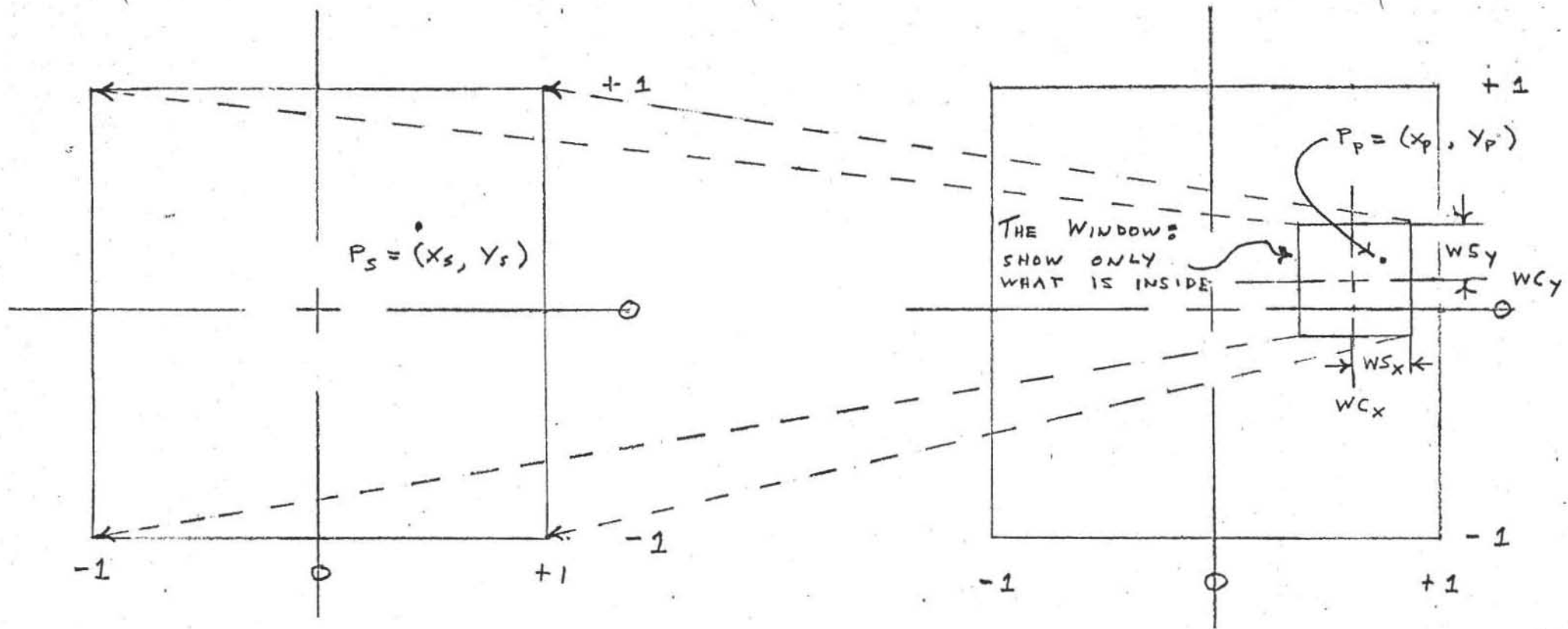
It is often convenient in a computer drawing to represent similar figures as instances of some common master figure. For example, the transistors of a circuit drawing might all be drawn from one master representation of a transistor. A program which can do this is like a rubber stamp. It enables its user to reproduce similar figures freely. The transformations represented in displaying an instance are very similar to the transformations involved in windowing. For example, suppose that a transistor symbol is to be drawn at a particular position on a drawing. Then, as shown in Figure 4A, there will be two sets of transformations. In the first transformation, a portion of a master page is reduced (or perhaps magnified) in scale and placed on the page to become a part of the drawing. A portion of the drawing will be displayed in a viewport on the scope by the windowing transformation. It follows, then, that a portion of the master picture may appear on the scope. When actually placing the lines and points of the master picture on the scope to represent a transistor, for example, it is convenient to use the same windowing program which is used otherwise. The parameters for the windowing job will, of course, be the concatenation of the two transformations involved. Instead of transforming the master picture into page coordinates and thence to scope coordinates, it is possible to transform directly from master coordinates to scope coordinates. (See Figure 4B)

If the instance to be drawn lies entirely outside of the window, of course, then there is no point in displaying any of the material in the

master drawing. If only a part of the instance area overlaps the window area, then only a part of the master picture material can possibly appear on the scope, and that only in a smaller "subviewport". These situations are shown in Figures 5, 6, and 7.

The use of instances in a picture implies the need to reduce two transformations to one. I have chosen to call this reduction process "Edging". The Edging task is to compute the single window W' and viewport V' which will provide the same result as would be provided by two transformations, one from the master picture to the page coordinate system and the other from the page coordinate system to the scope. The edging process is non-linear because the instance area may be larger than or smaller than or overlap with the window area. Depending on the relative size and location of the window and instance areas, the complete transformation may use a W' identical to that of the original master picture or some subset of the original master picture. Similarly the viewport V' may be the entire viewport or some subset of it.

The edging process is, of course, recursive. If the master picture is itself made up of instances, then the multiple transformation implied may still be reduced to a single transformation. In most practical cases, such multiple transformations result in smaller and smaller viewports and often, in fact, result in complete rejection of entire instances which lie entirely outside the window area.



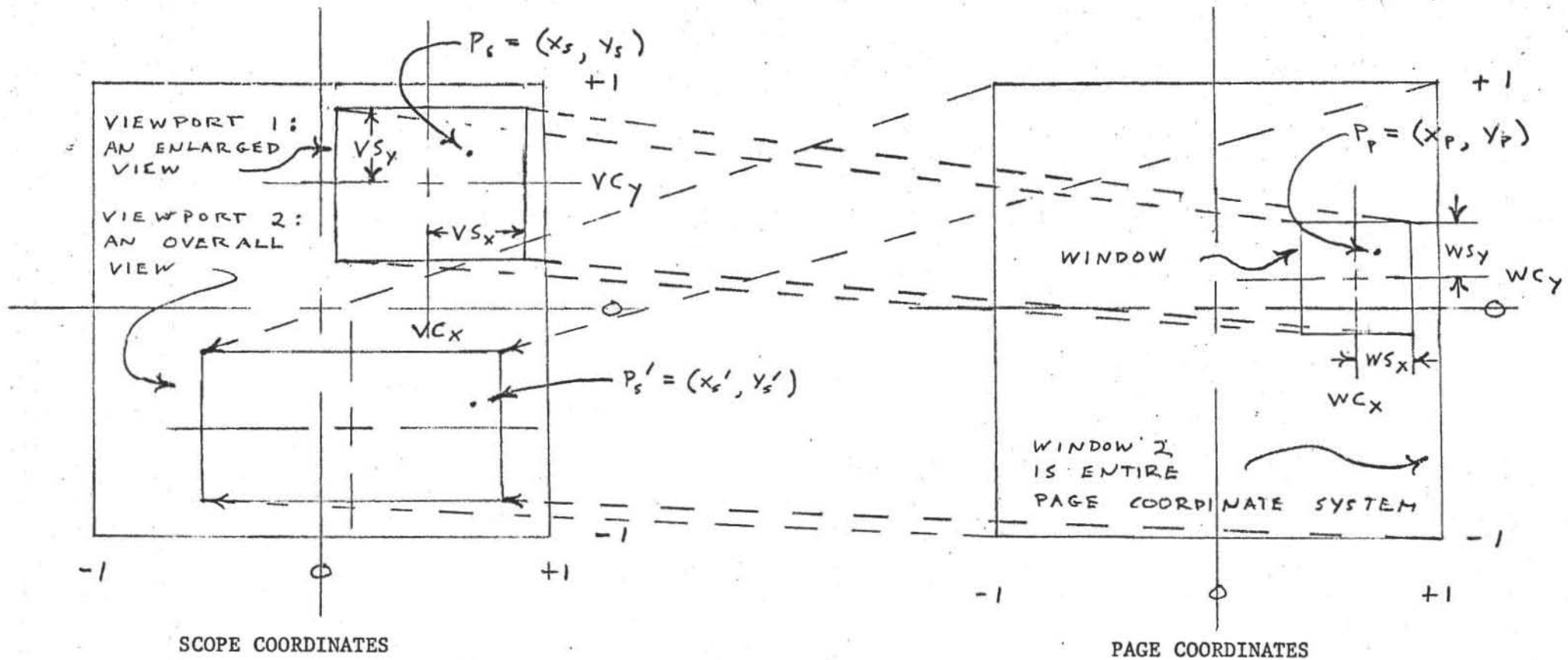
SCOPE COORDINATES

PAGE COORDINATES

$$x_s = \frac{x_p - wc_x}{ws_x}$$

$$y_s = \frac{y_p - wc_y}{ws_y}$$

FIGURE 1: Transformation From Page to Scope Coordinates

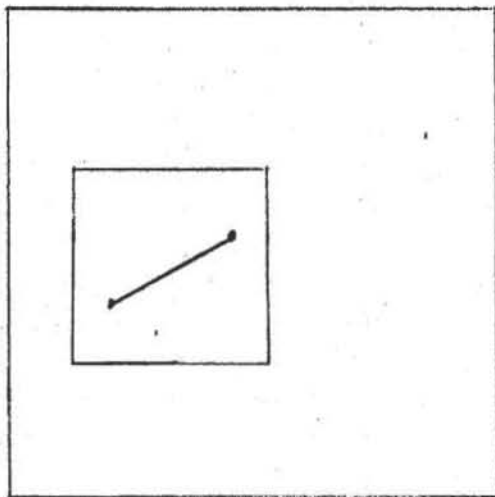


$$x_s = \frac{x_p - wc_x}{ws_x} VS_x + VC_x$$

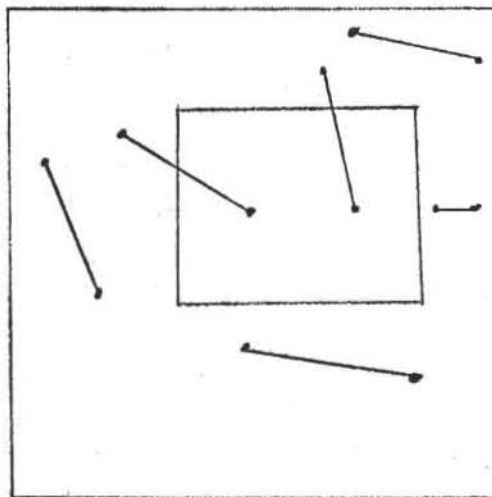
$$y_s = \frac{y_p - wc_y}{ws_y} VS_y + VC_y$$

FIGURE 2: TRANSFORMATION FROM PAGE TO SCOPE COORDINATES

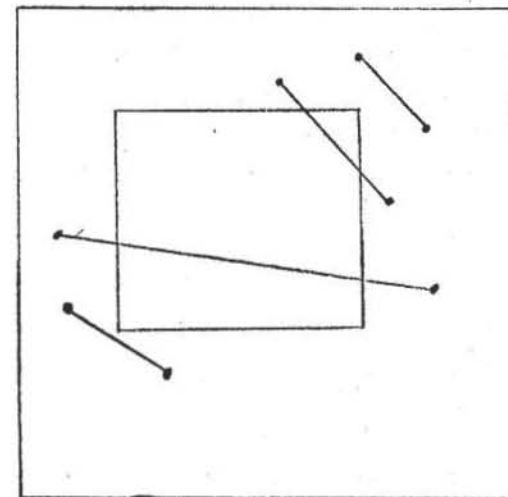
WITH VIEWPORT



A. Line inside Window



B. One end inside Window
Or Trivial Rejections



C. Doubtful Cases

FIGURE 3. LINES INSIDE AND OUTSIDE OF WINDOWS

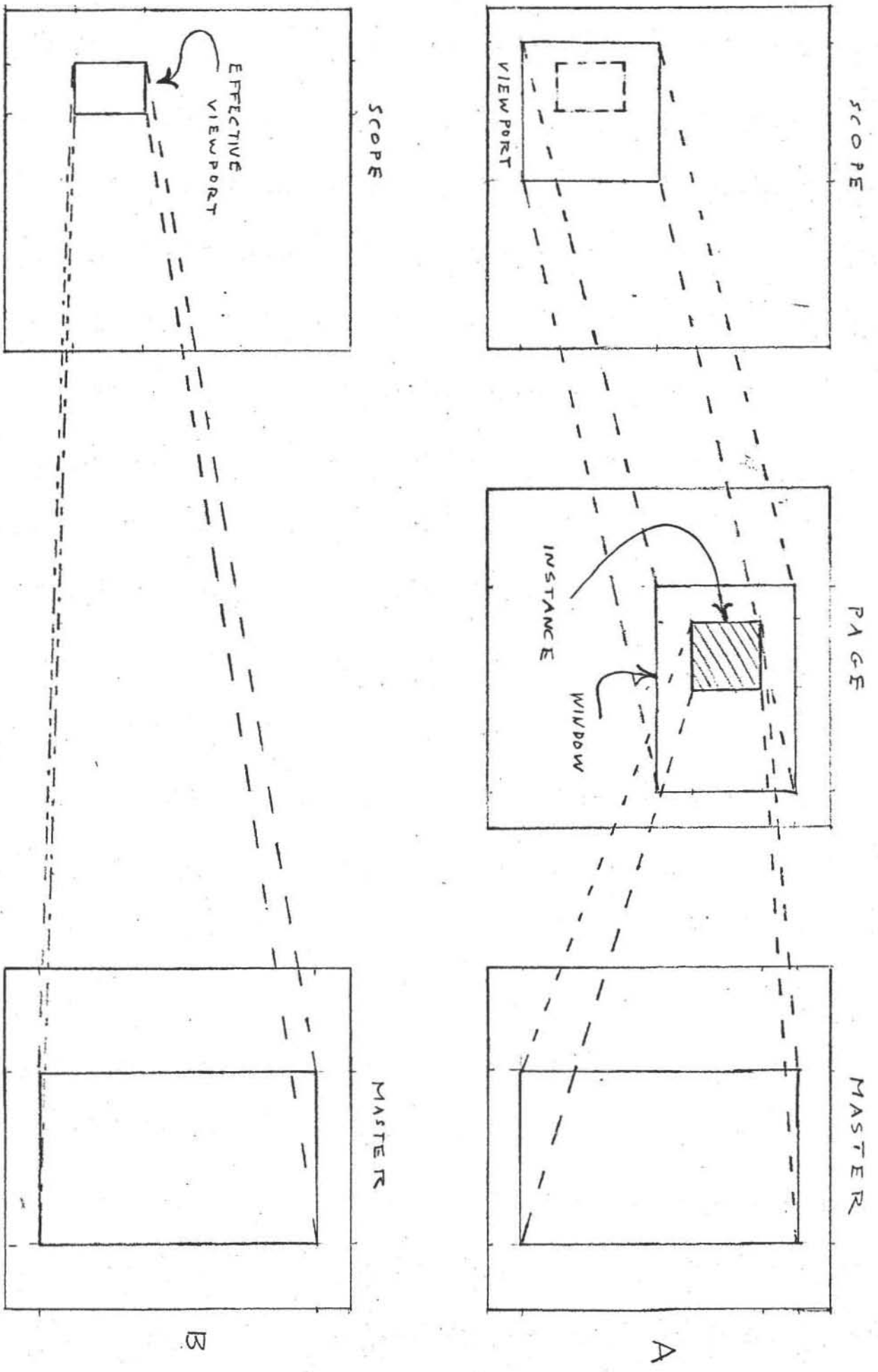


Figure 4 — THE EDGING PROCESS

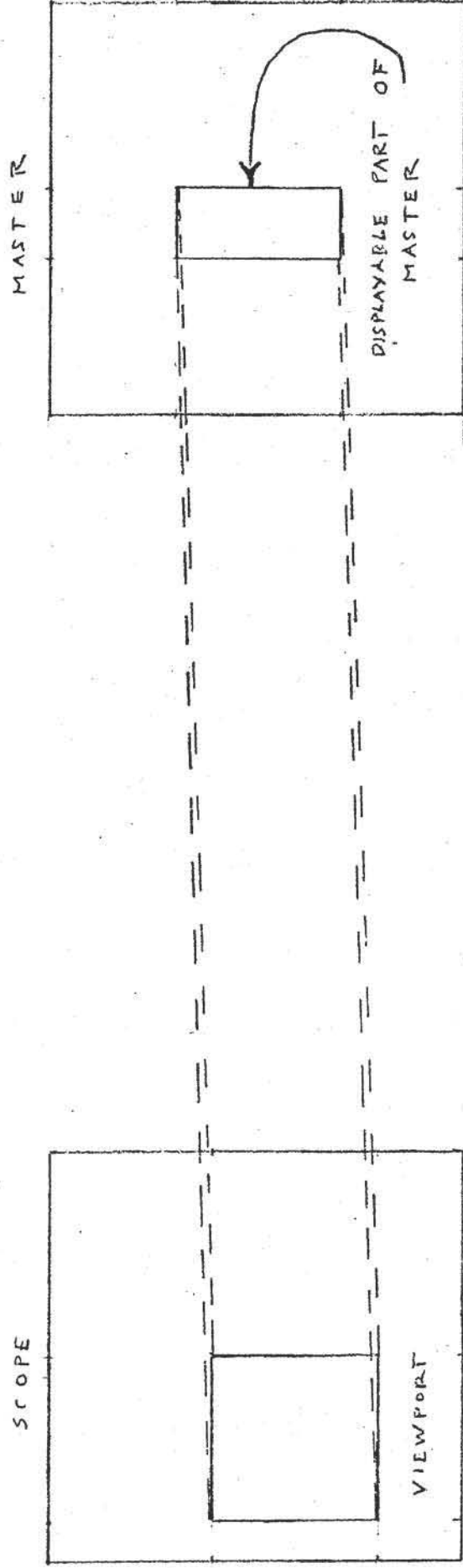
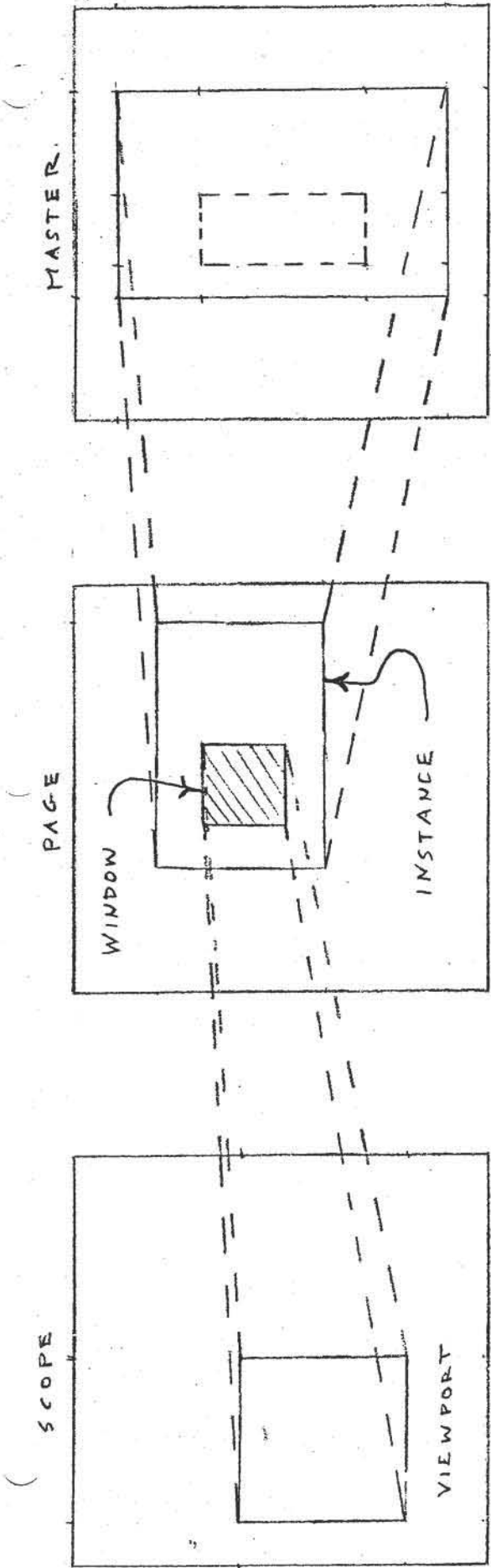


Figure 5 - Window Inside Instance

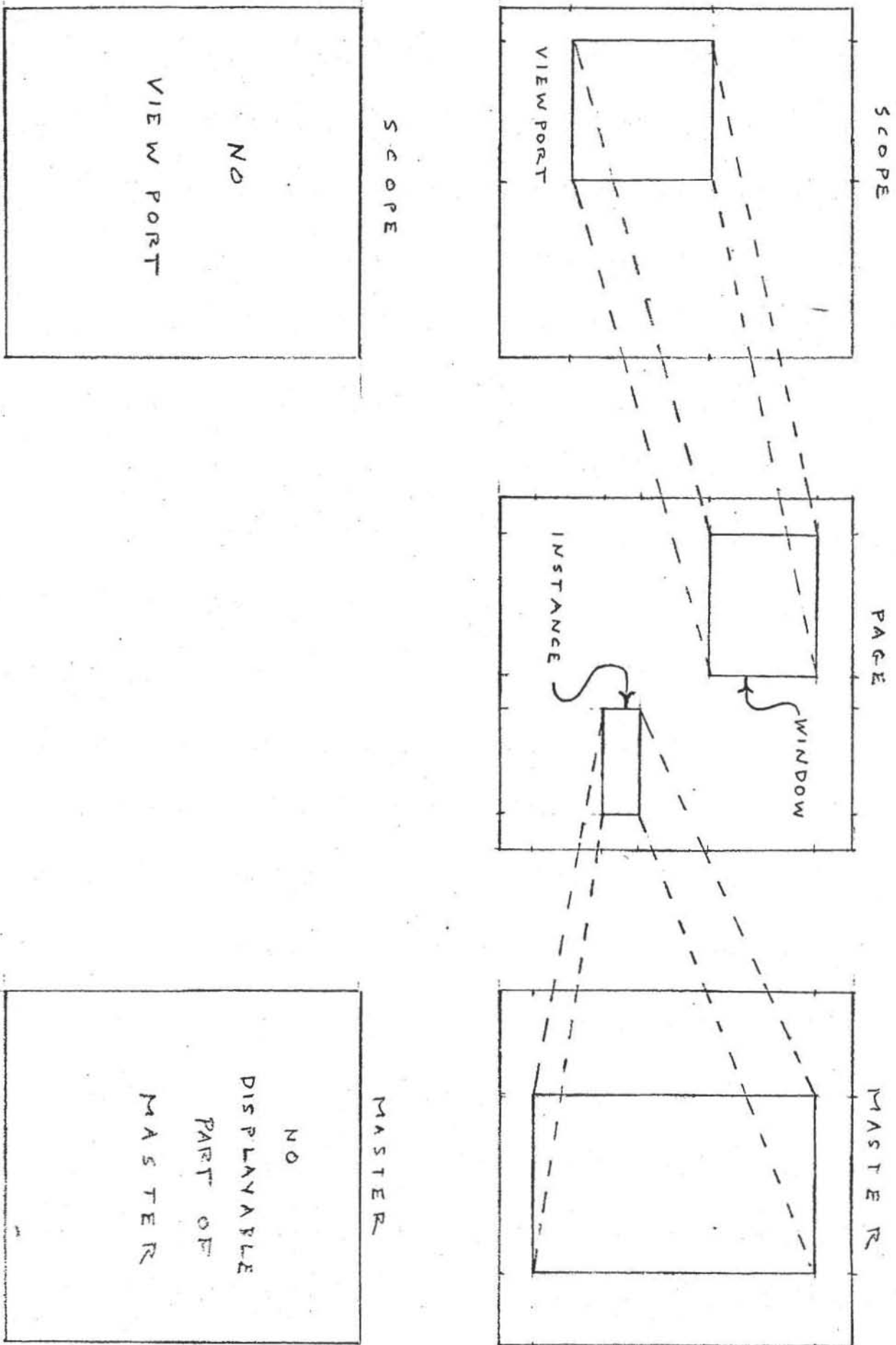


Figure 7 - Disjoint Window and Instance

CHAPTER FOUR

TRANSFORMATIONS FOR COMPUTER GRAPHICS

Matrix notation and the concept of homogeneous coordinates provide convenient tools for converting representations of objects in an internal coordinate system to the coordinate system used by the display. In particular, they readily provide the perspective transformation needed to display three-dimensional objects on the two-dimensional screen of the display.

This chapter contains examples of matrix operations being used for these purposes and introduces the use of homogeneous coordinates. It is, in fact, a summary of the paper, "Transformations and Matrices" by Professor Steven A. Coons, Appendix I, which should be consulted for the gory details. Future chapters will cover homogeneous coordinates more extensively in their use for parametric curve and surface drawing as related to the three-dimensional display processor.

Two Dimensions

Given a point (x,y) in two dimensions, we can transform it into another point (x',y') by the matrix multiplication

$$[x' \quad y'] = [x \quad y] \begin{bmatrix} a & b \\ c & d \end{bmatrix} = [ax+cy \quad bx+dy]$$

where we identify $x' = ax+cy$, $y' = bx+dy$. In particular, if we choose the proper forms for the matrix $T = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, we can completely characterize the types of transformation possible:

1. "shear" -- $T = \begin{bmatrix} 1 & d \\ c & 1 \end{bmatrix}$; a square is transformed to a parallelogram.

2. "scale" -- $T = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$; a square is transformed to a rectangle.

An important special case is $T = \begin{bmatrix} a & b \\ -b & a \end{bmatrix}$ and $a^2 + b^2 = 1$. Such a transformation causes a pure rotation by the angle θ in the XY plane where $a = \cos\theta$, $b = \sin\theta$.

There are two convenient ways of trying to understand the transformations described by matrices. One way is to see what happens to unit points such as the origin [100], [010], etc. The other method for dealing with transformations is to specify the transformed values of points of interest. The latter is a completely general way of determining the necessary transformation; the former is a quick way of observing its action.

By writing one vector (or point) below another, we can symbolically obtain the transform of several points at one time:

$$\begin{bmatrix} x_0 & y_0 \\ x_1 & y_1 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} x_0' & y_0' \\ x_1' & y_1' \\ \vdots & \vdots \\ x_n' & y_n' \end{bmatrix}$$

In particular, if we choose the unit points on the x and y axes, we have:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} x_0' & y_0' \\ x_1' & y_1' \end{bmatrix}$$

In this case, the values in the matrix itself are the coordinates of the transformed unit points!

On the other hand, let us try to find a matrix T which will transform the points $(x_0, y_0), (x_1, y_1)$ into the points $(x_0', y_0'), (x_1', y_1')$.

$$\begin{bmatrix} x_0 & y_0 \\ y_1 & y_1 \end{bmatrix} T = \begin{bmatrix} x_0' & y_0' \\ x_1' & y_1' \end{bmatrix}$$

if the matrix $\begin{bmatrix} x_0 & y_0 \\ x_1 & y_1 \end{bmatrix}$ has an inverse, which we will write $\begin{bmatrix} x_0 & y_0 \\ x_1 & y_1 \end{bmatrix}^{-1}$

we can pre-multiply by it to get:

$$\begin{bmatrix} x_0 & y_0 \\ y_1 & y_1 \end{bmatrix}^{-1} \begin{bmatrix} x_0 & y_0 \\ x_1 & x_1 \end{bmatrix} T = \begin{bmatrix} x_0 & y_0 \\ x_1 & y_1 \end{bmatrix} \begin{bmatrix} x_0' & y_0' \\ x_1' & y_1' \end{bmatrix}$$

$$T = \begin{bmatrix} x_0' & y_0' \\ x_1' & y_1' \end{bmatrix}^{-1} \begin{bmatrix} x_0 & y_0 \\ x_1 & y_1 \end{bmatrix}$$

Translation

The 2x2 transformation matrix used above cannot provide for simple translation. If we add a third row to the matrix, we can get translation to.

$$[x \ y \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a & b & 1 \end{bmatrix} = [x+a \ y+b \ 1] = [x' \ y' \ 1]$$

The addition of the third component allows us to think of two-dimensional transformations including translation in a single matrix formalism.

Three Dimensions

If we choose to represent points in 3-space by 3 component vectors, we can use a 3x3 matrix to represent a transformation in three dimensions:

$$[x \quad y \quad z] \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = [x' \quad y' \quad z']$$

It can be shown that any such transformation applied to the unit cube yields a parallelepiped. If we specify three points to be transformed and their resulting images, we have completely specified the transformation, just as two points serve in the two-dimensional case.

$$\begin{bmatrix} x_0 & y_0 & z_0 \\ x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \end{bmatrix} T = \begin{bmatrix} x_0' & y_0' & z_0' \\ x_1' & y_1' & z_1' \\ x_2' & y_2' & z_2' \end{bmatrix}$$

$$T = \begin{bmatrix} x_0 & y_0 & z_0 \\ x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \end{bmatrix}^{-1} \begin{bmatrix} x_0' & y_0' & z_0' \\ x_1' & y_1' & z_1' \\ x_2' & y_2' & z_2' \end{bmatrix}$$

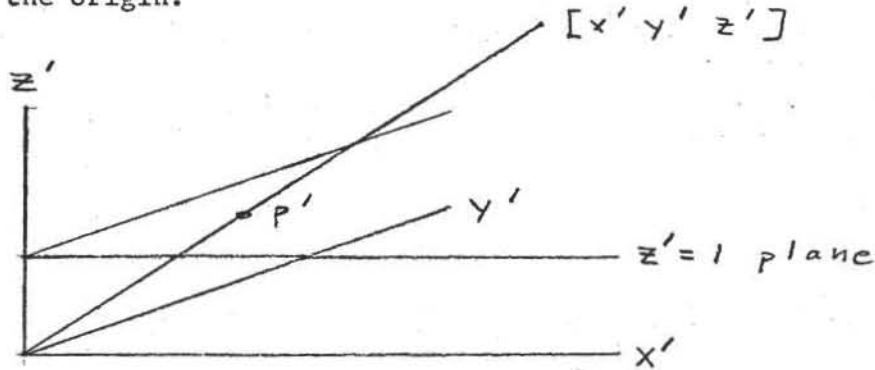
Homogeneous Coordinates

We can think of the transformation

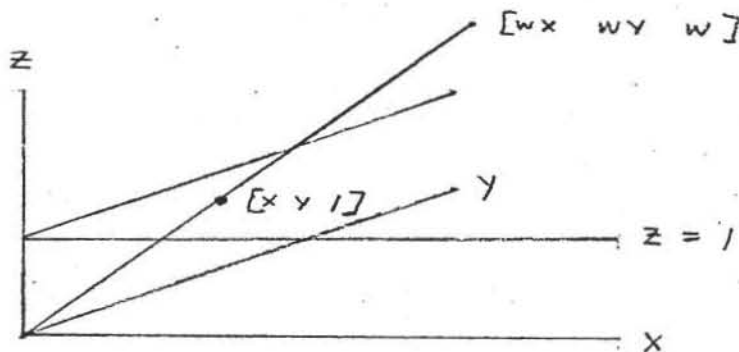
$$[x \quad y \quad 1] \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = [x' \quad y' \quad z']$$

as transforming a point (x,y) in the z=1 plane into the point (x',y',z') in 3 space. If we now divide this vector by its third component to

obtain a point $p' = (\frac{x'}{z'}, \frac{y'}{z'}, 1)$ we will have projected the point x', y', z' onto the point p' back on the $z=1$ plane by means of a ray through the origin.



Such operations lead us to define points in terms of homogeneous coordinates: The two-space point (x,y) is represented by a three component vector (wx, wy, w) where w is any non-zero quantity. Similarly, the triple (a,b,c) corresponds to the two-dimensional point $(\frac{a}{c}, \frac{b}{c})$. With such a representation, the two-dimensional point (x,y) can be regarded as the projection of any of the three-dimensional points (wx,wy,w) .



Line Equation

The ordinary equation of a line in two dimensions is $Ax+By+C = 0$; or, $[x \ y \ 1] \begin{bmatrix} A \\ B \\ C \end{bmatrix} = 0$

The column vector $\begin{bmatrix} A \\ B \\ C \end{bmatrix}$ thus can be used to represent the line. In homogeneous coordinates, we see that the equation is of the same form:

$$[wx \quad wy \quad w] \begin{bmatrix} A \\ B \\ C \end{bmatrix} = 0$$

If we transform a point (wx, wy, w) by the transformation T , then we can still write

$$[wx \quad wy \quad w] T T^{-1} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = 0$$

as the equation of the same line. Hence, a line,

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} \text{ transforms by the relation } T^{-1} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} A' \\ B' \\ C' \end{bmatrix}$$

Three-Dimensional Homogeneous Coordinates

In strict analogy with the above, we can write

$$[wx \quad wy \quad wz \quad w] [A] = [w's' \quad w'y' \quad w'z' \quad w']$$

where A is a 4×4 matrix. This yields a perspective transformation of three dimensions into three dimensions. If we ignore the third coordinate,

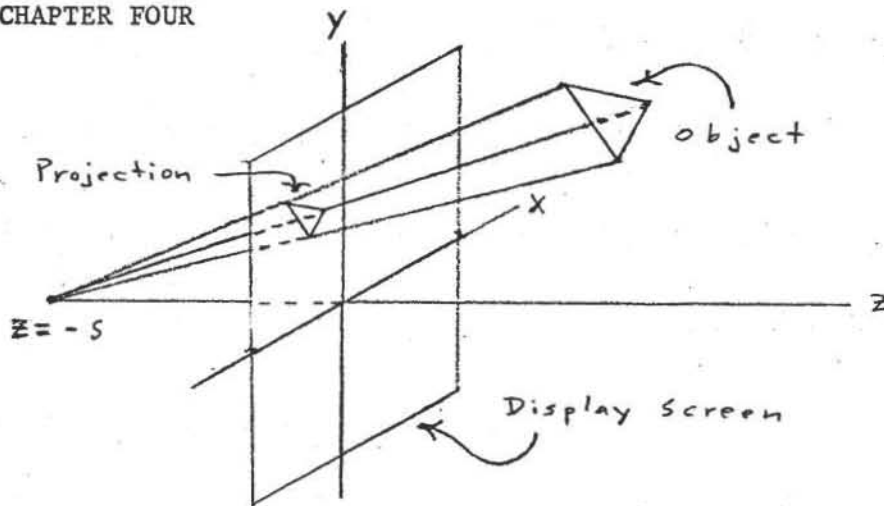
$(\frac{w'z'}{w'})$ and display the first two as a two-dimensional picture, $(\frac{w'x'}{w'}, \frac{w'y'}{w'})$

we obtain a perspective view of the object.

In particular, the transformation

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{s} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

gives the perspective view of an object displayed on a screen placed at $z=0$ as seen by an observer at $z=-s$:



In general, a 4x4 homogeneous transformation matrix can be written as

$$\begin{bmatrix} R & P \\ \text{---} & \text{---} \\ T & 1 \end{bmatrix}$$

where R is a rotation and scaling transformation, T is a translation, and P produces a perspective projection. Usually, we form such a matrix from several simple matrices -- R, T, or P -- by multiplying them together.

$$M = \begin{bmatrix} a & b & c & 0 \\ d & e & f & 0 \\ g & h & i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ j & k & L & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{s} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Such transformations can be viewed from the framework of photography or engineering drawing; their classification and properties are characterized in the appendix. Note that in interpreting the thin lens equation $\frac{1}{z} + \frac{1}{z'} = \frac{1}{f}$ as a transformation, a point (x, y, z) transforms into a point (x', y', z') where z' can cover quite a range; i.e., the lens needs to be focused. Also, objects at infinity - $W=0$ - may be imaged locally near the observer.

References

1. Noll, A. Michael, "A Computer Technique for Displaying N-Dimensional Hyper-objects", Communications of the ACM. August 1967, pp. 469-473
2. Maxwell, E.A., "General Homogeneous Coordinates in Space of Three Dimensions", Cambridge University Press, 1961
3. Roberts, L. G., "Homogeneous Matrix Representation and Manipulations of N-Dimensional Constructs", MIT Lincoln Laboratory Preprint MS-1405, May 1965

CHAPTER FIVE

REPRESENTATION OF DRAWING STRUCTURE

Most of the pictures produced by computer involve some highly structured subject matter. Were it not so, computer graphic representation would be far less useful; the representation of a beautiful art work by computer, for example, is not now considered useful even by artists. On the other hand, the representation of chemical molecules, engineering drawings, electrical drawings, and graph plots of orderly data are very useful. In this chapter, we will consider how some basic structures, such as points, lines and symbols, might be represented inside the computer for the production of pictures.

For simple pictorial notions, almost any representation will do, but for more complicated notions of relationships between parts of a picture, the particular form of storage that we choose will have a strong effect on the behavior of the resulting pictures. In an architectural context, for example, we might wish to indicate that all the windows on the front of a building should be the same size. How ought we to represent such a notion of similarity? In all useful cases that I have seen, it is the exceptions to such rules of similarity which are most important in the design. The columns in front of a building will have equal spacing EXCEPT where the center doorway comes out. All the flip flops of a register will be the same EXCEPT the first one and the last one. The choice of storage format that we make in any particular case will affect the things we can and cannot do with the resulting system.

Unfortunately, no one is sufficiently sophisticated in representing such abstract notions that we can provide the student with guidelines for how wisely to represent similarity, symmetry, connectivity or other such abstractions. In beginning any computer graphics problem, therefore, you should devote a good deal of effort to choosing an appropriate representation for the desired information.

Lines and Points

There are many possible representations for lines and points. For purely geometrical uses, a line might be considered as an infinitely long straight thing, and a point might be considered as a position in space. The numeric representations of lines and points can be made to be very similar if homogeneous coordinates are used: point coordinates can be represented by a row vector and line coordinates by a column vector. From this point of view, one might represent points and lines as very similar entities. In two dimensions, each would have three homogeneous coordinates.

POINT
GEORGE
X
Y
W

LINE
PETE
α
β
γ

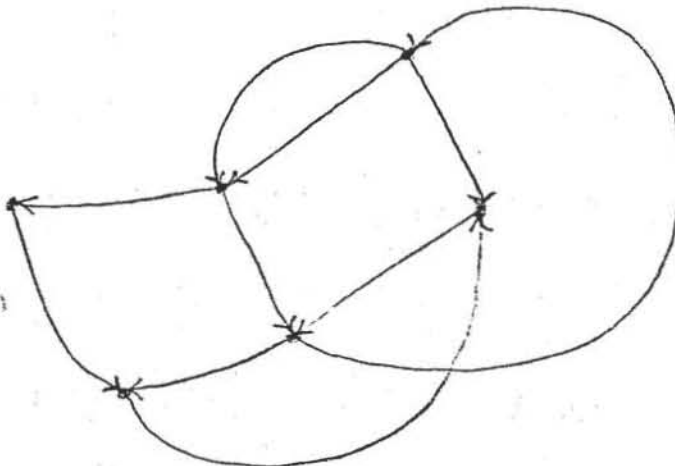
LINE EQUATION:

$$\alpha X + \beta Y + \gamma W = 0$$

Lines and points might be related to each other for mathematical applications by a representation in machine memory of the notion "lies on" or "passes through" (which really is only one notion viewed from a different point of view). The "lies on" or "passes through" notion associates points and lines.

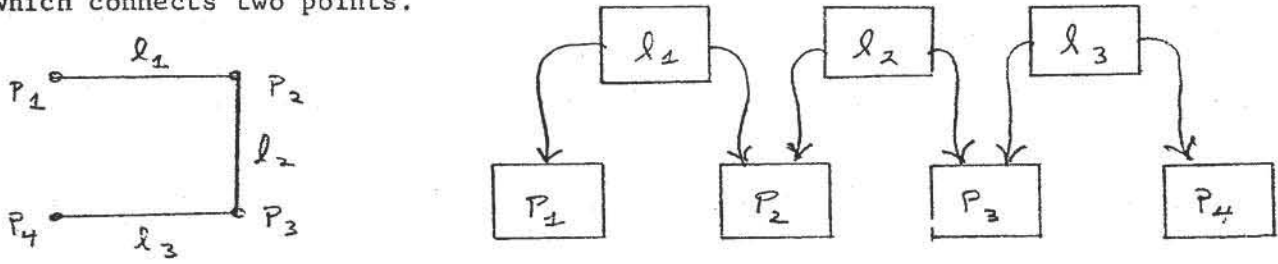
Unlike the representations of points and lines, the representation for the point-on-line relationship need not carry any specific data. On the other hand, it represents a condition upon the data contained in the point and the line. If the values in the point and line are to live up to the geometry intended by the point-on-line relationship, particularly when some of these values have been changed arbitrarily by the user, then a "demon" will have to be programmed which can adjust the coordinates of points and lines to satisfy all of the point-on-line conditions. The design and programming of such a demon may not be an easy matter. Such a representation of points, lines, and associations between them would be useful for doing projective geometry. Internally, lines would be of infinite length. For human consumption as much of the line would be displayed as would fit on the scope. Of course, the area shown on the scope would be variable so that a user might look at an entire figure or concentrate on a small section of it.

Suppose that instead of doing geometry we wish to deal with directed graphs, such as the one shown below.



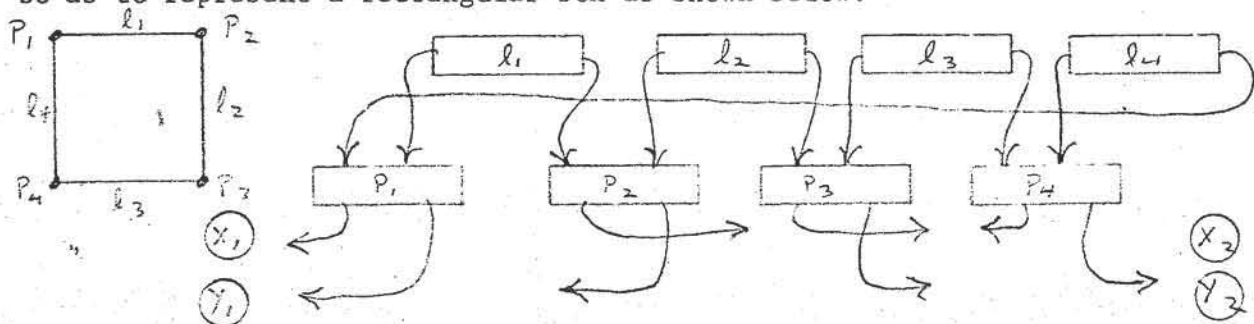
Such graphs might represent the behavior of a finite state machine. The important thing to represent here is the topology of the graph: Its geometry is of secondary importance. The topology of a picture is very simply represented if its points are named and the representations of its lines each contain the name of the point at which the line begins and the point at which it ends. Because the directedness of the line segments in such a graph would be important, appropriate tools would have to be provided with which a user could manipulate the topology and the orientation of the lines. For human consumption, of course, the lines of such a graph might not necessarily be drawn straight. For example, if two lines both run from point A to point B, it may be useful to represent each as a curved arc so that each may be seen. Additional information might be stored in the line segment representation to indicate how it is to be presented to the observer. Appropriate tools would also be needed to manipulate curvature of lines. Perhaps a special program to assist in presenting complex graphs might be written. Such a program might lay out a graph in such a way as to minimize the number of line crossings and maximize the symmetry of the presentation. Needless to say, the design and coding of such a program might not be a simple matter.

In ordinary engineering drawings, the lines with which one is concerned are undirected line segments. Such straight-line segments can be represented adequately by defining the locations of their end points. In a program for ordinary engineering drawings, then, one might represent coordinate data with points, but no coordinate data with lines. A line might be represented merely as an entity which connects two points.

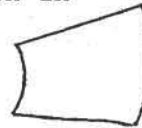


Several lines can of course terminate at the same endpoint, as shown in the figure. If a corner of such a box is moved, which can be done by merely changing the coordinates stored in that point, then both line segments attached to it will subsequently be seen in new locations. The structure of the representation says nothing about the length of a line or about the similarity of coordinates of the ends of a line which would make the line horizontal or vertical.

If one is especially interested in representing horizontal and vertical lines, the x or y coordinate information stored in the end points of a line would often be redundant. One might separate the coordinate information from the actual point blocks themselves so as to represent a rectangular box as shown below.



Changing one of the x coordinates will make the box wider or narrower and changing one of the y coordinates will make it taller or shorter. The structural representation, however, insists that the box be rectangular and alligned with the axis. A similar representation in polar coordinates might be used to represent circular wedges.



We have now seen three quite different examples of how points and lines might be represented. I hope to have convinced you by these examples that the choice of what to represent, to say nothing of how to represent it, depends strongly upon the application you have in mind. The proper early care in choosing what to represent will save a world of grief later on.

Representing Curves

Entities more complicated than straight-line segments will need richer representation. A circular arc for instance, might be represented as depending on the positions of three points: a center point, a point at which the arc starts, and a point at which the arc terminates. In addition, of course, one would need a single bit to represent the sense of the arc. Unfortunately, such a representation implies that the two points at the ends of the arc be equidistant from the center. Suppose they are not. What should then be drawn for human consumption? Here you have many choices. The display program for circles should make some picture which is suitable for the job at hand. In "Sketchpad",

I chose to draw the arc using the radius defined by the startpoint and only the angle defined by its endpoint as shown below.

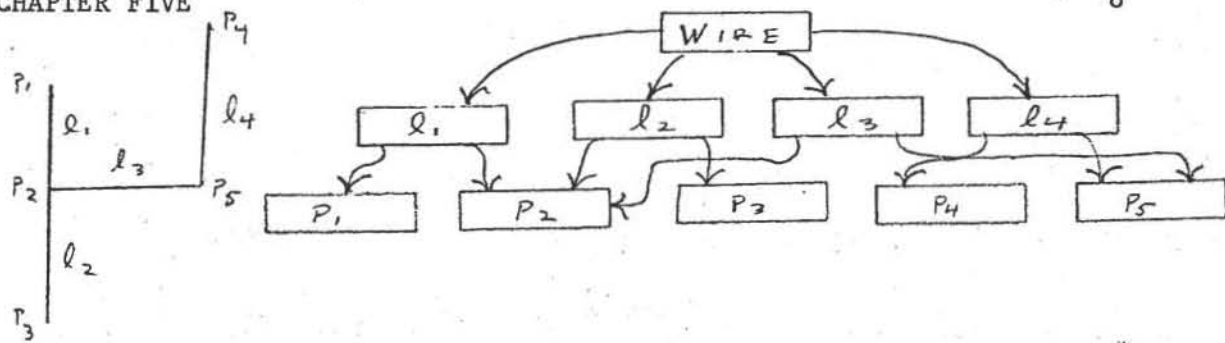


Alternatively, one could draw the arc with the mean radius indicated or the maximum radius indicated or the minimum, or any other arbitrary choice which is suitable to the problem at hand.

Each thing represented in memory should have some presentation form for human consumption. This may be as simple as a dot for a point, or as complex as the curved lines in the direct graph example. Such display forms can be related in many complex ways to the data presented. For example, a display program might let one choose between the mechanical layout view or electrical circuit view of an integrated circuit represented in memory.

Collections of Things

Collections of lines in a drawing are often of significance. In an electrical drawing, for example, a collection of lines may constitute a wire, whereas another collection of lines may constitute a component. In representing a wire one can either choose to store it in memory as a single thing, or one could choose to superimpose the collection idea on top of the elementary form in which lines are represented. For example, one might represent a wire as shown below:



Given such a representation in memory, one can readily program such features as the ability to erase an entire wire or to brighten it on the display by pointing to any part of it. Given such a representation, however, it is not so easy to insure that the parts of a wire follow the horizontal and vertical lines commonly used in electrical diagrams. One might instead choose to represent a wire, not as a collection of lines, but as an entity in its own right, an entity with N segments, each of which has a horizontal or vertical position and duration in the other coordinate.

Regardless of what representation is used for a wire, there may be electrical or mechanical properties OF THE WIRE which should be represented with it. For example, the wire may have an electrical potential or may be open wire, twisted pair, or coax. Such properties are best represented as properties of the wire rather than of its individual segments.

Instances

The most common collections of lines on a drawing are those used to make symbols. Symbols used in electrical, mechanical and mathematical drawings are usually geometrically similar. That is, although the size, position, and orientation of a particular kind of symbol may vary from place to place, it is usually of exactly the same shape. Because of this similarity, it is possible to represent a particular instance of a symbol by reference to a picture which defines its shape.

Thus, for example, the drawing of a flip flop might contain two references to the transistor symbol definition. Each such reference might indicate a size and position in which the symbol is to be drawn. The actual content of the symbol need not be stored again and again with each reference to it. I have chosen to call such a "rubber stamp" reference to a picture an "instance". I will speak of an "instance of a transistor" which means, of course, a specific reference to the transistor symbol definition. One can "move an instance", which means to change the parameters in the instance block so that the transistor symbol appears in a different position on the picture. One can "delete an instance" which means to eradicate from memory the particular reference involved. To "delete an instance" does not mean to delete the definition to which the instance makes reference. I will call the definition to which an instance makes reference its "Master Picture". Several instances may make reference to the same master picture. One wants to consider very carefully what parameters one includes in the definition of an instance. In the "Sketchpad" program, for example, an instance contained four parameters x , y , $A \sin \alpha$, $A \cos \alpha$, where A is the size of the instance and α is its angle of rotation. This choice of parameters did not permit mirror-image instances, which caused a great deal of difficulty in making transistor drawings, because transistor symbols come both right and left handed.

On the other hand, a system designed specifically with electrical drawings in mind, could be useful even with only a single size of instance available, because all similar symbols on an electrical drawing will be the same size. In such a system, an instance need only contain as parameters x and y position and three additional bits, two to indicate orientation, and one to indicate mirror image symmetry. In some other circumstance, an instance might, perhaps, need to contain separate scale factors for horizontal and vertical dimensions. With an appropriate choice of parameters, all rectangles could be represented as instances of a square. Roberts' block-drawing program, for example¹, treated all parallelepipeds as instances of a cube and all triangular wedges as instances of an equiangular triangular wedge, all tetrahedra as instances of an equilateral tetrahedron, etc.

In order to display an instance for human consumption, a computer graphics program must display all of the lines and points and curves and other displayable material which appear in its master picture but in the reduced size and changed orientation indicated by the parameters of the instance. As far as the display is concerned, an instance is a sort of subroutine which says "go display all of the stuff which appears in that master picture, but with these parameters". The parameters given may affect different parts of the master picture in different ways. For example, text in the master picture might be displayed in horizontal orientation regardless of the orientation of the rest of the material in the picture.

¹ Roberts, Lawrence G., "Machine Perception of Three-Dimensional Solids", Doctoral Thesis, MIT, June 1963

If a character generator is used, the text might be displayed in one of the available sizes convenient to the character generator even though this size might not be exactly correct. In "Sketchpad", digits which represented distances were modified in content so that when displayed in an instance they assumed an appropriate value. For example, length labels on the sides of a 3, 4, 5 triangle would in instances of it always be in 3, 4, 5 proportion but with specific values appropriate to the size of the instance.

The expansion of instances is, of course, a recursive procedure. Thus, for example, if the master picture of an instance contains instances of some other master, then those subinstances must be expanded as a part of the expansion of the master. A flip flop symbol which contains transistor symbols may require expansion two layers deep. The route by which the instance expanding program has entered a complex instance structure is conveniently kept in a push down stack.

As soon as one allows recursive expansion of instances, of course, one has the possibility of tangling a drawing. Suppose, for example, one puts into a drawing an instance of itself. Such a drawing used to appear on Morton Salt labels. They showed a girl who carries another Morton Salt package smaller than the first, but upside down so that salt is draining out of it. On the smaller package appears a girl who is carrying a still smaller package upside down so that salt is draining out of it, and so on. Or for another example, consider Claude Shannon's bus: Shannon remodeled the bus for camping, but before embarking on the full-size bus, he built a model bus to help plan the layout. When the large

When the large bus was complete, he put the model bus in it - which required, of course, that in the model there appeared a model bus, etc.

There are several ways of handling this paradox. One can either forbid circularity of instances or use some rule to truncate their expansion. One possible rule is to continue the expansion until no additional significant information is put into the drawing, i.e. until the subinstances become so small that their detail is lost, or so large that none of their lines pass through the scope. Another possible rule is to terminate the expansion as soon as the circularity is detected as I did in "Sketchpad". In my system, one could see only the first instance of self, but no successive subinstances.

As pointed out in the chapter on windowing, instance expansion really involves reproducing some section of the master picture in some subsection of the page coordinates of the picture in which it is to appear. This in turn implies that material from some portion of the master picture may appear in some subviewport on the scope. Of course, if the instance lies entirely outside the "window", the part of the page that can be seen on the scope, then the instance need not be expanded. The ability to reject an entire instance as lying outside the present window without having to expand it to test its parts individually can result in major time savings in displaying a complex drawing.

During the expansion of an instance, one would like to provide a single layer windowing function which would go directly from master

coordinates to the scope. This is possible only if the windowing transformation available is matched to or richer than the transformation carried in an instance. If, for example, an instance can call for mirror imaging, and the windowing algorithm cannot, mirrored instances will have to be expanded by special means. The page-to-scope transformation should be rich enough so that when concatenated with the master-to-page transformation implied in an instance, the resulting transformation is no more complicated than the page-to-scope transformation can handle.

In three dimensions, then, we are faced with the need to represent surfaces and solids. Surfaces can be represented as associating several lines, or preferably as occupying the region interior to a set of points. The simplest plane surface, analogous to the line segment in two dimensions, is the plane triangle interior to three points. Such a triangle is fully defined by the points and like the 2-line segment needs no further data in its representation. Evans and his group at the University of Utah are using such triangles as the basic elements in making perspective views of solid objects with hidden parts removed.

In a later chapter we will see how curved surfaces may be represented as blending four curves which define their boundary. Topologically, of course, the surface so represented ties together the four curves. We are now involved in some experiments, moreover, where the data representing the nature of the surface patch should be stored. Perhaps the data relevant to a curve should be stored with it, and only the additional data about the surface (how it bulges, etc.) should be stored with the surface. I suspect, with no proof, that a representation of a triangular surface patch might be handier. Unfortunately, however, our simple mathematical formulation for the surface uses two orthogonal parameters and so rectangular patches naturally result. A clean formulation of triangular surface patches is needed. Such a formulation would let us coat a sphere, a task now considered difficult.

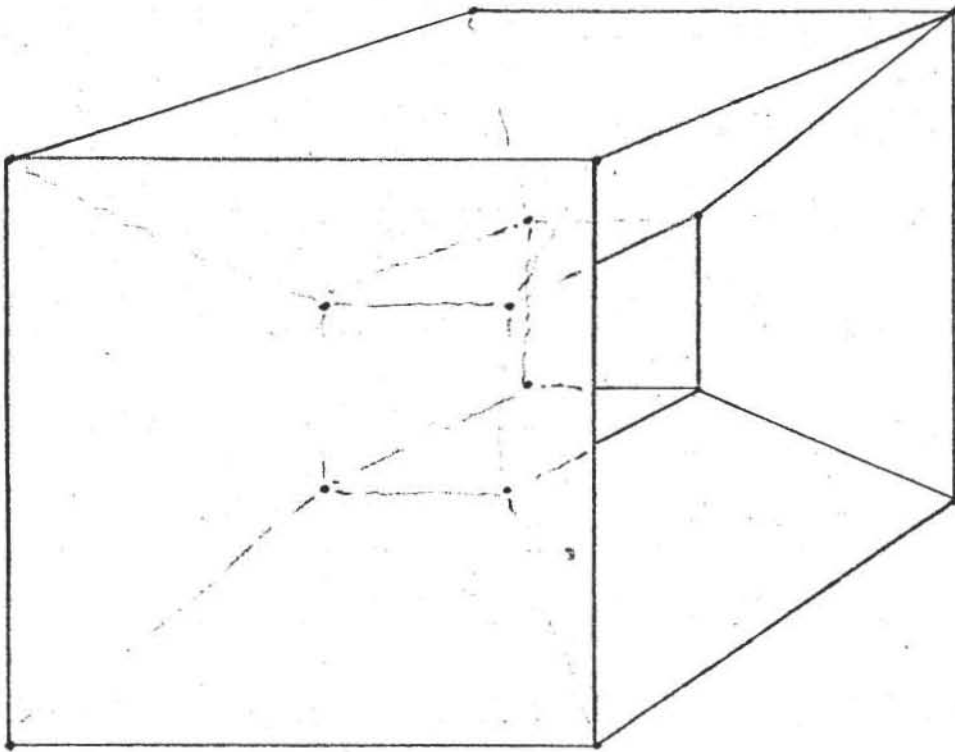


FIGURE 2 B

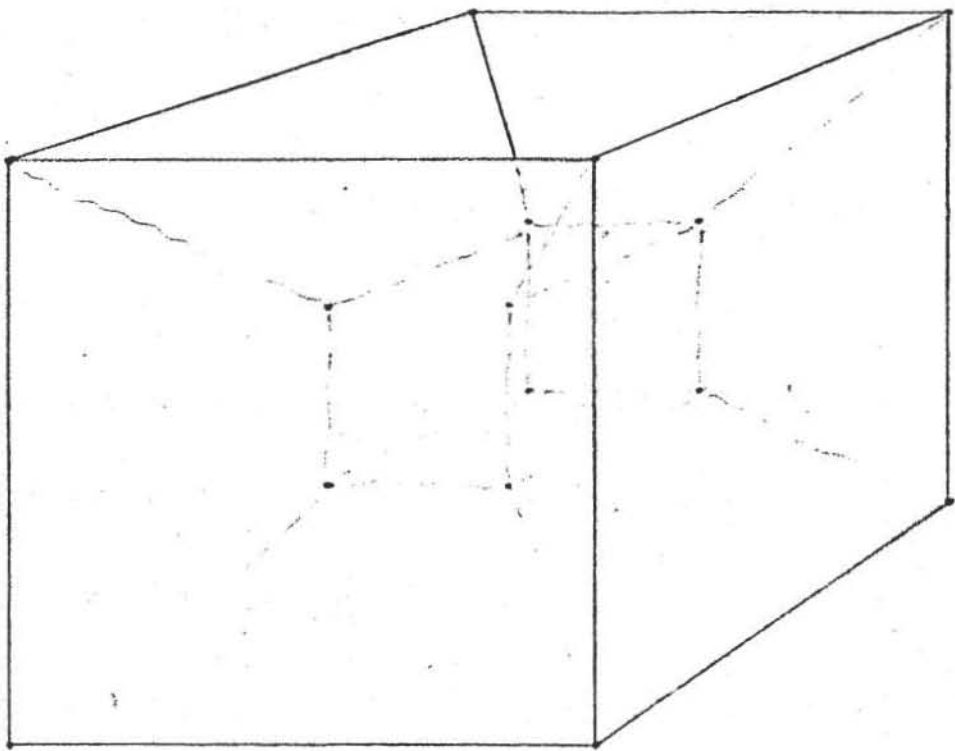


FIGURE 2 C

CHAPTER SIX

CONIC - DRAWING DISPLAYS

Larry Roberts, then of the MIT Lincoln Laboratory, devised a scheme for drawing conic sections on a computer display¹. Roberts' scheme utilizes multiplying digital to analog converters to generate deflection voltages appropriate for tracing the conic sections. The scheme has subsequently been built by Howard Blatt also of the MIT Lincoln Laboratory² and is currently in operation on the TX-2 computer. Because Roberts' scheme involves nine multiplying D to A converters and the generation of voltages which are quadratic in time, it appears at first glance to be very complicated. It is the purpose of this chapter to show how to use the conic-generating hardware to draw the curve of your choice.

Roberts' hardware implements the function

$$\begin{bmatrix} t^2 & t & 1 \end{bmatrix} \begin{bmatrix} \alpha & a & d \\ \beta & b & e \\ \delta & c & f \end{bmatrix} = \begin{bmatrix} wx & wy & w \end{bmatrix} \quad (1)$$

His hardware is arranged to plot the values of

$$x = \frac{wx}{w} \quad \text{and} \quad y = \frac{wy}{w}$$

as t ranges from 0 to 1. Notice that the x and y plots are the ratios of quadratic expressions of the parameter t .

$$x = \frac{wx}{w} = \frac{\alpha t^2 + \beta t + \delta}{\delta t^2 + \epsilon t + f} \quad y = \frac{wy}{w} = \frac{a t^2 + b t + c}{\delta t^2 + \epsilon t + f} \quad (2)$$

¹Roberts, Lawrence G., "Conic Display Generator Using Multiplying Digital-Analog Converters", IEEE Transactions on Electronic Computers, Volume EC-16, Number 3, June 1967

²Blatt, Howard, "Conic Display Generator Using Multiplying Digital/Analog Decoders", Presented at the Fall Joint Computer Conference, Anaheim, California, Fa-1, 1967

It can be shown that all curves represented in this manner are conic sections, and that all conic sections can be represented by this form.

The way that Roberts' hardware works is as follows: The matrix terms a through f are stored in digital registers. The bits of these registers are used to select weighting registers in an amplifier circuit such that each of the entries in the matrix can multiply an analog voltage by the digital fraction which it stores. The output of the three multiplying digital to analog converters in each column are added together. The analog inputs to the multiplying digital to analog converters for the three rows are provided with signals of the form kt^2 , kt and k respectively. A feedback circuit controls k from the third column of the matrix in such a way that the output of the third row is always one, i.e. that the expression

$$dkt^2 + ekt + fk = w = 1$$

This insures that the output of the other two columns are the real values of X and Y rather than w_x and w_y .

Conceptually then, Roberts' curve drawer is very simple. In practice, to make the feedback circuit stable proved to be a very difficult task. The feedback circuit is required to do the division implied in equations (2). Indeed the feedback loop is stable only for values of the parameters in the following ranges:

$$0.25 \leq \frac{w}{k} \leq 3$$

$$-1 < d < 1$$

$$-1 < e < 1$$

$$\frac{1}{2} \leq f \leq 1$$

Care must be taken to see that the parameters provided to specify the curve produce values within this range.

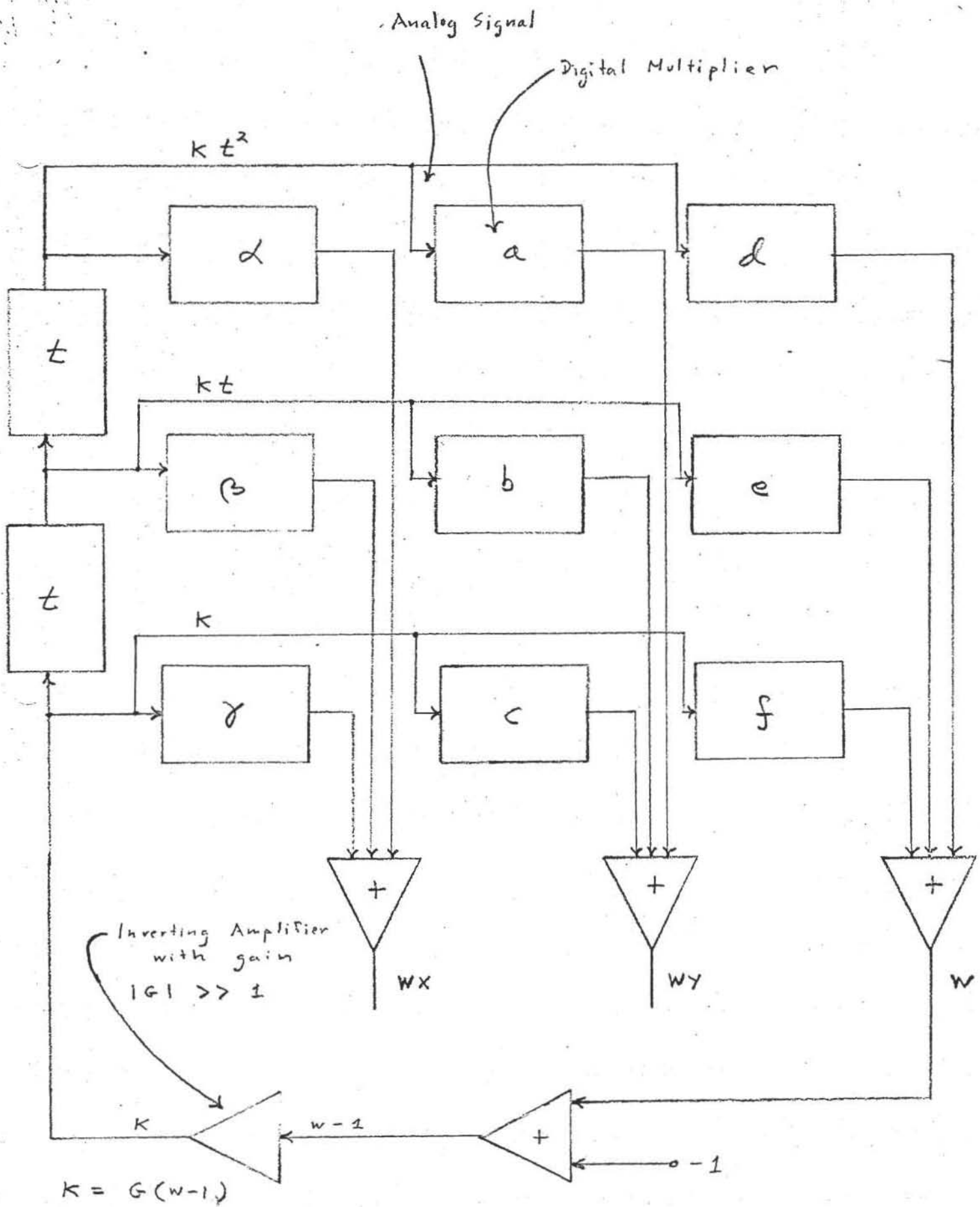


Figure 1
Conic Generator

How The Curve Is Drawn

Let us consider the geometry of the curve

$$[x \ y \ 1] = [t^2 \ t \ 1] \quad (3)$$

in the range $0 \leq t \leq 1$. Obviously all points of this curve lie on the $w=1$ plane.

Also obviously, the curve begins at the origin and ends at the point $[111]$.

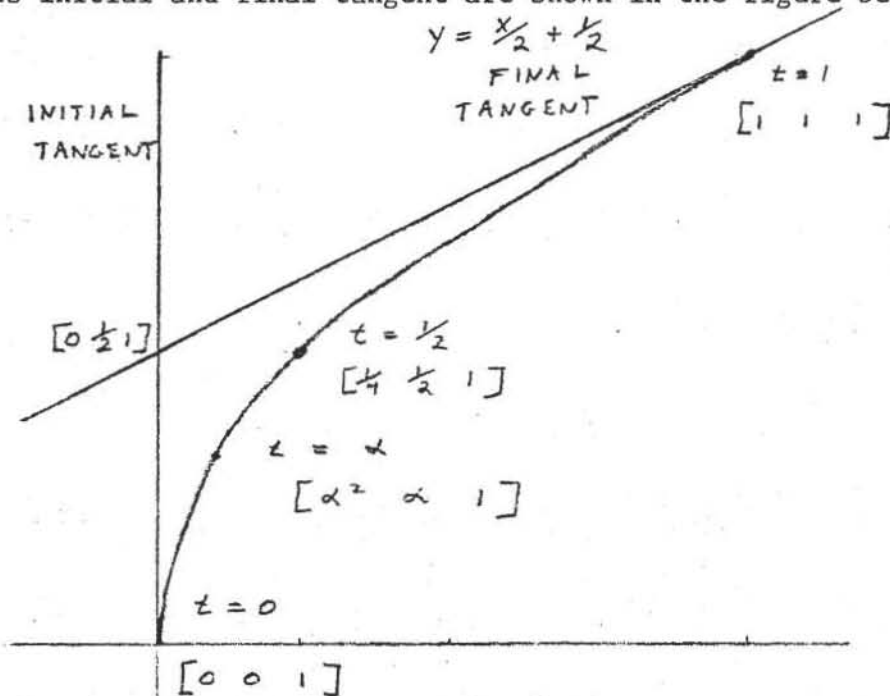
The derivative of the curve with respect to t is

$$\left[\frac{dx}{dt} \ \frac{dy}{dt} \ \frac{dw}{dt} \right] = [2t \ t \ 0] \quad (4)$$

From this we can see that the slope of the tangent to the curve (which is the y derivative divided by the x derivative) is given by $\frac{dy}{dx} = \frac{1}{2t}$. From this it follows that the equation of the tangent line to the curve is

$$y = \frac{x}{2t} + \frac{t}{2} \quad (5)$$

At the beginning of the curve ($t=0$) the curve is at the origin and has a vertical tangent. At the end of the curve ($t=1$) the curve is at the point $[111]$ and its tangent has slope of $1/2$ and y intercept of $1/2$. The curve and its initial and final tangent are shown in the figure below.



When $t=\alpha$, the curve passes through the point $[\alpha^2 \quad \alpha \quad 1]$.

Now suppose that we wish to map this canonic conic curve into some other curve. We can define the shape of the other curve by specifying its endpoint and the point at which its endpoint tangents intersect. We can further specify the curve we wish to draw by specifying the location of some point on it, including the value that the parameter t should have when the curve passes through that point. We might, for example, specify the position of the midpoint ($t=\frac{1}{2}$) of the curve. How do we derive the nine values for a matrix T which will transform the $[t^2 \quad t \quad 1]$ curve into the desired conic section?

If we call the start point, tangent intersection, and end point which define the conic V_o , V_t , and V_1 respectively, then it follows that

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & \frac{1}{2} & 1 \\ 0 & 0 & 1 \end{bmatrix} T = \begin{bmatrix} V_1 \\ V_t \\ V_o \end{bmatrix} \quad (6)$$

because the canonic end, intersection, and start points must be so mapped.

Now the inverse of

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & \frac{1}{2} & 1 \\ 0 & 0 & 1 \end{bmatrix} \text{ is } \begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

which I will call M . Thus,

$$T = M \begin{bmatrix} V_1 \\ V_t \\ V_o \end{bmatrix} \quad (7)$$

which defines one possible transformation T .

Actually, the vectors V_o , V_t and V_1 can be scaled by any arbitrary scale factors which I will call w_o , w_t and w_1 . $V_o = [x_o \quad y_o \quad 1]$ is the same point as $w_o V_o = [w_o x_o \quad w_o y_o \quad w_o]$. Therefore we can say

$$T = M \begin{bmatrix} w_1 V_1 \\ w_t V_t \\ w_o V_o \end{bmatrix}$$

and we can get many possible transformations depending on our choice of the w 's.

We can choose w_o , w_t and w_1 so that the resulting curve hits some point V_c at some specified time $t=\alpha$. For this to be true,

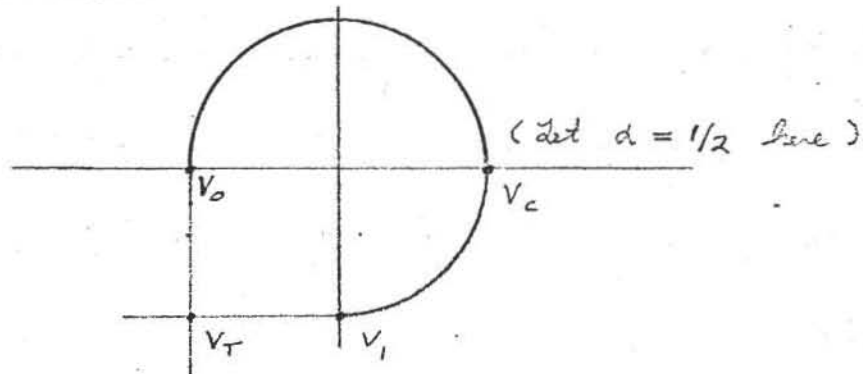
$$[\alpha^2 \quad \alpha \quad 1][T] = V_c \quad (8)$$

$$[\alpha^2 \quad \alpha \quad 1][M] \begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_t & 0 \\ 0 & 0 & w_o \end{bmatrix} \begin{bmatrix} V_1 \\ V_t \\ V_o \end{bmatrix} = V_c$$

$$[\alpha^2, 2\alpha(1-\alpha), (1-\alpha)^2] \begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_t & 0 \\ 0 & 0 & w_o \end{bmatrix} = [x_c \quad y_c \quad 1] \begin{bmatrix} V_1 \\ V_t \\ V_o \end{bmatrix}^{-1} \quad (9)$$

Equation (9) is just a vector equation in the three unknowns, w_1 , w_t and w_o . Given a choice of α , it will tell us the values of w_1 , w_t and w_o to use.

For example, suppose we want to produce three quarters of a unit circle about the origin.



For this case, equation (9) becomes

$$\begin{bmatrix} \frac{1}{4}w_1 & \frac{1}{2}w_t & \frac{1}{4}w_o \end{bmatrix} = [1 \ 0 \ 1] \begin{bmatrix} 0 & -1 & 1 \\ -1 & -1 & 1 \\ -1 & 0 & 1 \end{bmatrix}^{-1}$$

$$\begin{bmatrix} w_1 & 2w_t & w_o \end{bmatrix} = [4 \ 0 \ 4] \begin{bmatrix} 1 & -1 & 0 \\ 0 & -1 & 1 \\ 1 & -1 & 1 \end{bmatrix} = [8 \ -8 \ 4]$$

$$w_1 = 8$$

$$w_t = -4$$

$$w_o = 4$$

$$T = \begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -8 & 8 \\ 4 & 4 & -4 \\ -4 & 0 & 4 \end{bmatrix} = \begin{bmatrix} -12 & -16 & 20 \\ 16 & 8 & -16 \\ -4 & 0 & 4 \end{bmatrix}$$

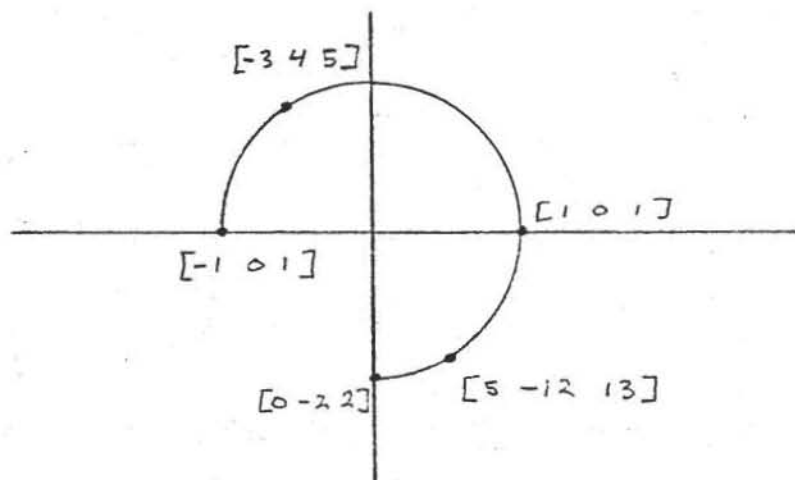
Since we can drop the scale factor for all of T , we will divide out a factor of 4 and use

$$T = \begin{bmatrix} -3 & -4 & 5 \\ 4 & 2 & -4 \\ -1 & 0 & 1 \end{bmatrix}$$

Now let us check where some typical values come out

t	t^2	t	1	wt^2	wt	w				
0	0	0	1	0	0	1	\times	$\begin{bmatrix} -3 & -4 & 5 \\ 4 & 2 & -4 \\ -1 & 0 & 1 \end{bmatrix}$	$=$	$\begin{bmatrix} -1 & 0 & 1 \\ -3 & 4 & 5 \\ 1 & 0 & 1 \\ 5 & -12 & 13 \\ 0 & -2 & 2 \end{bmatrix}$
$\frac{1}{4}$	$\frac{1}{16}$	$\frac{1}{4}$	1	1	4	16				
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{2}$	1	1	2	4				
$\frac{3}{4}$	$\frac{9}{16}$	$\frac{3}{4}$	1	9	12	16				
1	1	1	1	1	1	1				

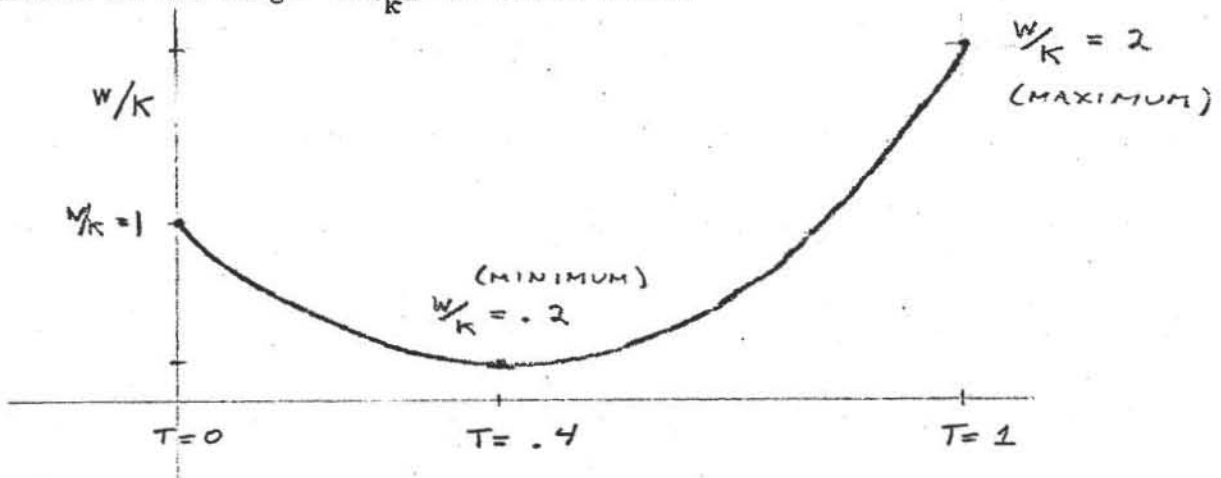
All of which are at once recognizable as being on the desired circle!



Sectioning

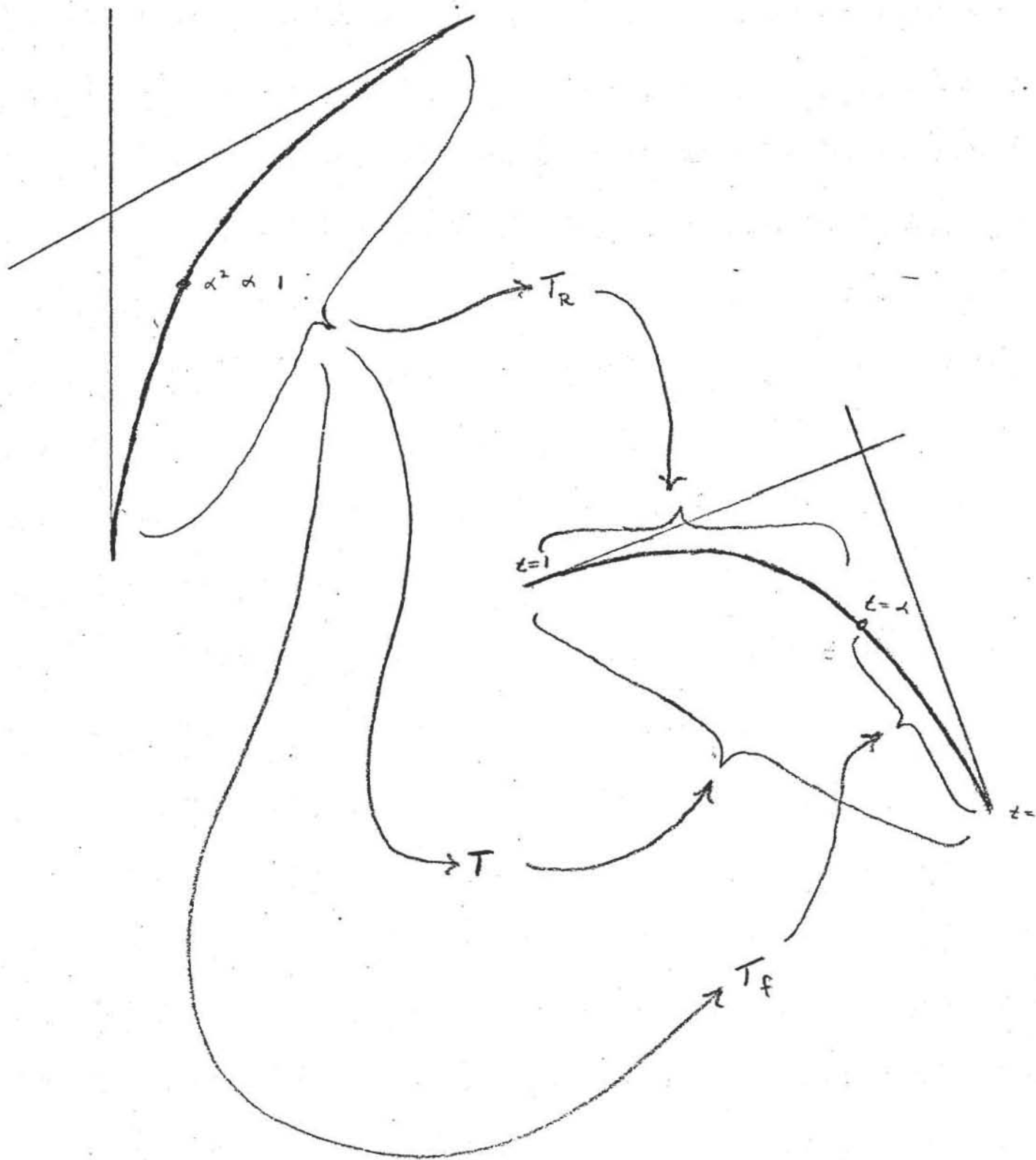
Suppose that we have a transformation T which transforms the $[t^2 \ t \ 1]$ curve into the curve that we wish to see. Suppose however that the transformation T is unacceptable to the hardware because, for example, it produces values of $\frac{w}{k}$ unacceptably large or small.

Such is the case in the example for which $\frac{w}{k} = (5t^2 - 4t + 1)$. $\frac{w}{k}$ varies in the range $.2 \leq \frac{w}{k} \leq 2$ as shown below.



It would be nice to draw the desired curve in two segments using two separate settings of the conic generator, one for each segment. Can we derive transformations appropriate to each of the two segments from the transformation for the full curve?

Suppose that the curve is to be divided at the point where the parameter t has the value α (typically $\alpha = \frac{1}{2}$). We can generate a new matrix T_α which draws the first part of the curve by transforming the $[t^2 \ t \ 1]$ curve into the part of the target curve for which t runs from zero to α . We can generate another matrix $T_{1-\alpha}$ which transforms the $[t^2 \ t \ 1]$ curve into the rest of the target curve with parameter values from α to one. These transformations are shown symbolically in the Figure on the following page.



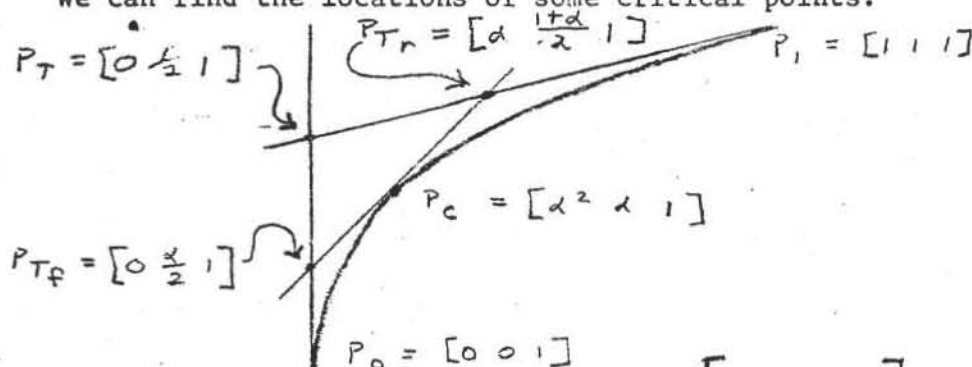
We can derive the transformations T_f and T_r by first transforming the full $[t^2 \ t \ 1]$ curve into a part of itself, and then transforming that part into the corresponding part of the target curve by the transformation T . In other words,

$$T_f = A_f T \quad \text{and} \quad T_r = A_r T,$$

where the matrices A_f and A_r may be derived as the appropriate transformation of points in the original space. Thus, for example, the transformation A_f should map the $[t^2 \ t \ 1]$ curve into the portion of itself running from the origin to the point $[\alpha^2 \ \alpha \ 1]$. The transformation A_r should map the $[t^2 \ t \ 1]$ curve into the portion of itself running from the point $[\alpha^2 \ \alpha \ 1]$ to $[1 \ 1 \ 1]$. By referring to equation (5) for the tangent to the $[t^2 \ t \ 1]$ curve at the point where $t=\alpha$

$$y = \frac{x}{2\alpha} + \frac{\alpha}{2}$$

We can find the locations of some critical points.



Obviously A_f maps $[P_1 \ P_t \ P_0]$ into $[P_c \ P_{tf} \ P_0]$ and A_r maps $[P_1 \ P_t \ P_0]$ into $[P_1 \ P_{tr} \ P_c]$. Thus we can write:

$$A_f = \begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha^2 & \alpha & 1 \\ 0 & \frac{\alpha}{2} & 1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha^2 & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (10)$$

$$A_r = \begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ \alpha & \frac{1+\alpha}{2} & 1 \\ \alpha^2 & \alpha & 1 \end{bmatrix} = \begin{bmatrix} (1-\alpha)^2 & 0 & 0 \\ 2\alpha(1-\alpha) & 1-\alpha & 0 \\ \alpha^2 & \alpha & 1 \end{bmatrix} \quad (11)$$

To check these results, let us see how A_f and A_r affect the $[t^2 \ t \ 1]$ curve.

$$[t^2 \ t \ 1] \begin{bmatrix} \alpha^2 & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} = [\alpha^2 t^2 \quad \alpha t \quad 1]$$

which is obviously the first portion of itself.

$$[t^2 \ t \ 1] \begin{bmatrix} (1-\alpha)^2 & 0 & 0 \\ 2\alpha(1-\alpha) & 1-\alpha & 0 \\ \alpha^2 & \alpha & 1 \end{bmatrix} = [(1-\alpha)^2 t^2 + 2\alpha(1-\alpha)t + \alpha^2 \quad (1-\alpha)t + \alpha \quad 1]$$

$$= [\{(1-\alpha)t + \alpha\}^2 \quad \{(1-\alpha)t + \alpha\} \quad 1]$$

again obviously a portion of the curve.

Let us now section the three quarter circle of our example. We will section it at the place where $t = \frac{1}{2}$.

$$T_f = A_f T = \begin{bmatrix} \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -3 & -4 & 5 \\ 4 & 2 & -4 \\ -1 & 0 & 1 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} -3 & -4 & 5 \\ 8 & 4 & -8 \\ -4 & 0 & 4 \end{bmatrix}$$

which maps the $[t^2 \ t \ 1]$ curve into the upper semicircle, and

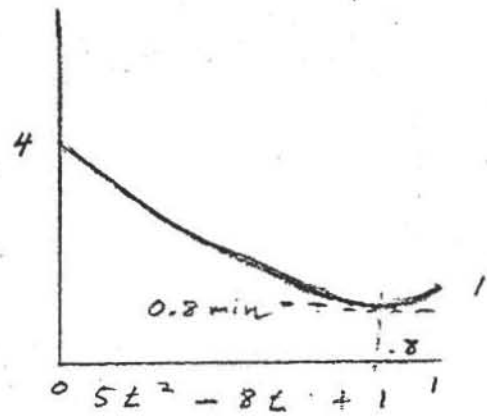
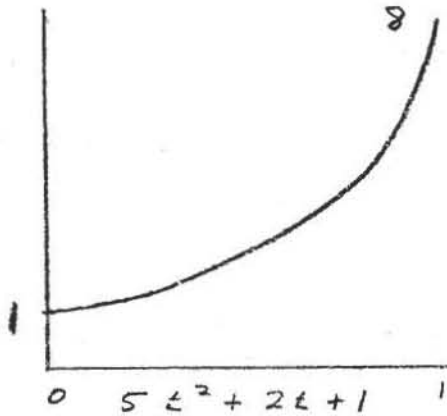
$$T_r = A \cdot T_r = \begin{bmatrix} \frac{1}{4} & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{4} & \frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} -3 & 4 & 5 \\ 4 & 2 & -4 \\ -1 & 0 & 1 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} -3 & -4 & 5 \\ 2 & -4 & 2 \\ -3 & 0 & 1 \end{bmatrix}$$

which maps it into the quarter circle in the fourth quadrant. For these two transformations, the $\frac{w}{k}$ ratios are

$$\frac{w}{k} = (5t^2 - 8t + 4) \quad \text{and}$$

$$\frac{w}{k} = (5t^2 + 2t + 1)$$

which look like:



Thus the range of $\frac{w}{k}$ ratios is now even worse!

The Squishing Transformation.

Let us now find a transform which will map the $[t^2 \quad t \quad 1]$ curve into itself but move the point where $t=\alpha$ to the point where $t=1-\alpha$. To do this, we need to find an appropriate set of w 's.

$$[\alpha^2 \quad \alpha \quad 1] \begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_t & 0 \\ 0 & 0 & w_o \end{bmatrix} = [(1-\alpha)^2, 1-\alpha, 1] \begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \alpha^2 w_1 & 2\alpha(1-\alpha)w_t & (1-\alpha)^2 w_o \end{bmatrix} = \begin{bmatrix} (1-\alpha)^2 & 2\alpha(1-\alpha) & \alpha^2 \\ \alpha^2 & 2\alpha & 1 \end{bmatrix}$$

$$w_1 = \left(\frac{1-\alpha}{\alpha}\right)^2 = \gamma$$

$$w_t = 1$$

$$w_o = \left(\frac{\alpha}{1-\alpha}\right)^2 = \frac{1}{\gamma}$$

$$A_s = \begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \gamma & \gamma & \gamma \\ 0 & \frac{1}{2} & 1 \\ 0 & 0 & \frac{1}{\gamma} \end{bmatrix} = \begin{bmatrix} \gamma & \gamma-1 & \gamma + \frac{1}{\gamma} - 2 \\ 0 & 1 & 2\left(1 - \frac{1}{\gamma}\right) \\ 0 & 0 & \frac{1}{\gamma} \end{bmatrix}$$

$$= \begin{bmatrix} \gamma & \gamma-1 & \frac{\gamma-2\gamma+1}{\gamma} \\ 0 & 1 & 2\left(\frac{\gamma-1}{\gamma}\right) \\ 0 & 0 & \frac{1}{\gamma} \end{bmatrix} = \begin{bmatrix} \gamma & \gamma-1 & \frac{(\gamma-1)^2}{\gamma} \\ 0 & 1 & 2\left(\frac{\gamma-1}{\gamma}\right) \\ 0 & 0 & \frac{1}{\gamma} \end{bmatrix}$$

Now let us check that A_s really maps $[t^2 \quad t \quad 1]$ into itself

$$\begin{aligned} [t^2 \quad t \quad 1] A_s &= \left[\gamma t^2, \gamma t^2(\gamma-1) + t, \frac{(\gamma-1)^2}{\gamma} + 2\frac{(\gamma-1)}{\gamma} + \frac{1}{\gamma} \right] \\ &= \frac{1}{\gamma} \left[\gamma^2 t^2, \gamma t (t(\gamma-1) + 1), (t(\gamma-1) + 1)^2 \right] \\ &= \frac{1}{\gamma} (t(\gamma-1) + 1)^2 \left[\frac{\gamma^2 t^2}{(t(\gamma-1) + 1)^2}, \frac{\gamma t}{t(\gamma-1) + 1}, 1 \right] \end{aligned}$$

which is easily recognized as being of the form $[u^2 \quad u \quad 1]$ where

$$u = \frac{\gamma t}{t(\gamma-1)+1}$$

We can factor out a γ from the A_S matrix to make it slightly clearer and use

$$A_S = \begin{bmatrix} \gamma^2 & \gamma(\gamma-1) & (\gamma-1)^2 \\ 0 & \gamma & 2(\gamma-1) \\ 0 & 0 & 1 \end{bmatrix}$$

The objective of using the A_S transform is to make a T matrix which has minimum variation in $\frac{w}{k}$. Such matrices have the property that their upper right corner entry and the one just below it are equal and opposite in sign. Given a matrix T of the form

$$T = \begin{bmatrix} \cdot & \cdot & a \\ \cdot & \cdot & b \\ \cdot & \cdot & 1 \end{bmatrix}$$

The product

$$A_S T = \begin{bmatrix} \cdot & \cdot & a\gamma^2 + b\gamma(\gamma-1) + (\gamma-1)^2 \\ \cdot & \cdot & b\gamma + 2(\gamma-1) \\ \cdot & \cdot & 1 \end{bmatrix}$$

From which, by an "obvious" reduction it follows that

$$\gamma^2 = \frac{1}{a + b + 1}$$

If we attempt to apply the A's transform to the 3/4 circle, we find

$$\gamma^2 = \frac{1}{5-4+1} = \frac{1}{2},$$

~~Application~~ Application to the half circle yields

$$\gamma^2 = \frac{1}{\frac{5}{4} - \frac{8}{4} + 1} = 4$$

and application to the quarter circle yields

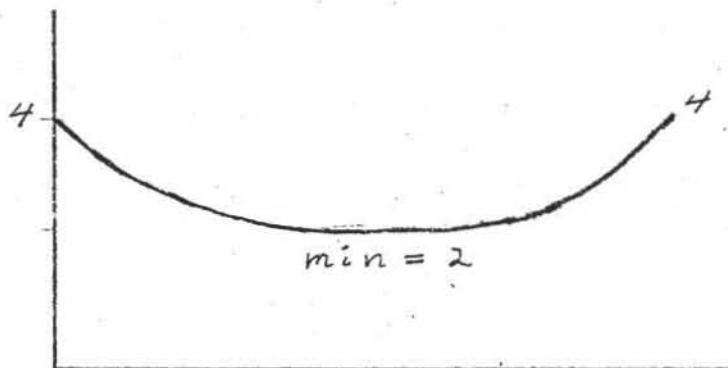
$$\gamma^2 = \frac{1}{5+2+1} = \frac{1}{8}$$

which we can use to find a minimum-w-variation equivalent transform.

For the half circle, for example,

$$A_s T = \begin{bmatrix} 4 & 2 & 1 \\ 0 & 2 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -3 & -4 & 5 \\ 8 & 4 & -8 \\ -4 & 0 & 4 \end{bmatrix} = \begin{bmatrix} 0 & -8 & 8 \\ 8 & 8 & -8 \\ -4 & 0 & 4 \end{bmatrix}$$

This matrix provides for equal starting and ending $\frac{w}{k}$ values as shown below.



CHAPTER SEVEN

"STYLUS INPUT DEVICES FOR COMPUTER GRAPHICS

As Told In Part By

Thomas G. Stockham, Jr.

While cathode ray tube display devices provide an adequate output medium for computer graphics, they do not themselves provide for input of graphic material. A variety of stylus devices are available, however, which enable their user to draw information directly into the computer. It is the purpose of this chapter to describe the properties of stylus input devices and how they are used.

There are two basic types of stylus input devices: pointing devices and positioning devices. The pointing and positioning functions are quite different. An ideal stylus input device would contain both pointing and positioning capability. Unfortunately, no stylus device inherently provides both capabilities, although most stylus input devices can be made to behave as if they had both properties. Both pointing and positioning devices are usually used in conjunction with cathode ray tube displays. Pointing devices enable their user to point out a particular item already on the display picture: an item such as a line or a character for example. Positioning devices, on the other hand, enable their user only to indicate the coordinates of a single point which the computer can most easily use in positioning objects in the picture. Pointing devices in effect say "this thing" whereas positioning devices in effect say "here".

Fairly sophisticated software is required to obtain pointing information from a positioning device. Since the computer knows only the coordinates delivered by the device, a program must compare those

coordinates with every displayed object to discover the closest match. Although the comparison task is not difficult for points and straight lines, it is rather more difficult for curves, and quite time-consuming in any case. It is generally more practical to provide pointing hardware (See Page 12) than to expect the program to search the picture for a position match.

Similarly, a sophisticated program is required to obtain position information from a pointing device. In an appendix to this chapter, we consider such "tracking" programs in detail. Some pointing devices are provided with special hardware to provide the position function through automatic tracking. Other pointing devices are equipped with special hardware which makes it virtually impossible to obtain the position information, even with a very sophisticated program.

POSITIONING DEVICES

There are many schemes by which the computer can be informed of the position of a stylus held in a human operator's hand. Of these, the most prevalent and versatile are the so-called "Rand Tablet" and the "Voltage Gradient Stylus". Both the Rand Tablet and the Voltage Gradient Stylus use electrical fields to detect the stylus position. It is also possible to use magnetic fields, sound, light, and mechanical techniques to sense the position of the stylus. In all cases, the function of the equipment is merely to indicate the coordinates of the tip of the stylus to the computer at regular intervals of time. How this coordinate information is used is up to the computer program.

The Rand Tablet

The Rand Tablet¹, also known as the "Teager table" is a simple digital device for detecting the position of a stylus. In the surface of the tablet are located 1,024 vertical lines and 1,024 horizontal lines. Each line is made of copper about 3 thousandths of an inch wide and one thousandth of an inch thick. The lines are spaced about one one-hundredth of an inch apart so that the active area of the tablet is typically ten inches by ten inches. The horizontal lines are separated from the vertical ones by a thin sheet of milar.

The individual vertical and horizontal wires are brought out at the edge of the board to coding devices. In the device developed by Ellis and Sibley at Rand, the coding is obtained through capacitive coupling between the extended wires and a special pattern of copper plates etched on the reverse side of the thin milar sheet. The pattern of capacitor plates is so arranged that ten pairs of pulses placed sequentially in

¹ Ellis, T. O. and Davis, M. R., "The Rand Tablet: A Man-Machine Graphical Communication Device", Proceedings of the Fall Joint Computer Conference, Vol. 26, pp. 325-31, 1964.

time are coupled differently into each of the individual wires. In one particular wire, for instance, all of the pulse will be positive followed by negative. In other wires, certain of the pulse pairs will be positive-negative whereas others will be negative-positive. The coding used is a "Gray-code" scheme so that the sequence of pulses in each wire is unique and the sequence of pulses in two adjacent wires differs only in one pulse position. Teager's implementation of the device is similar in principle but uses a different technique for coupling the pulses into the wires.

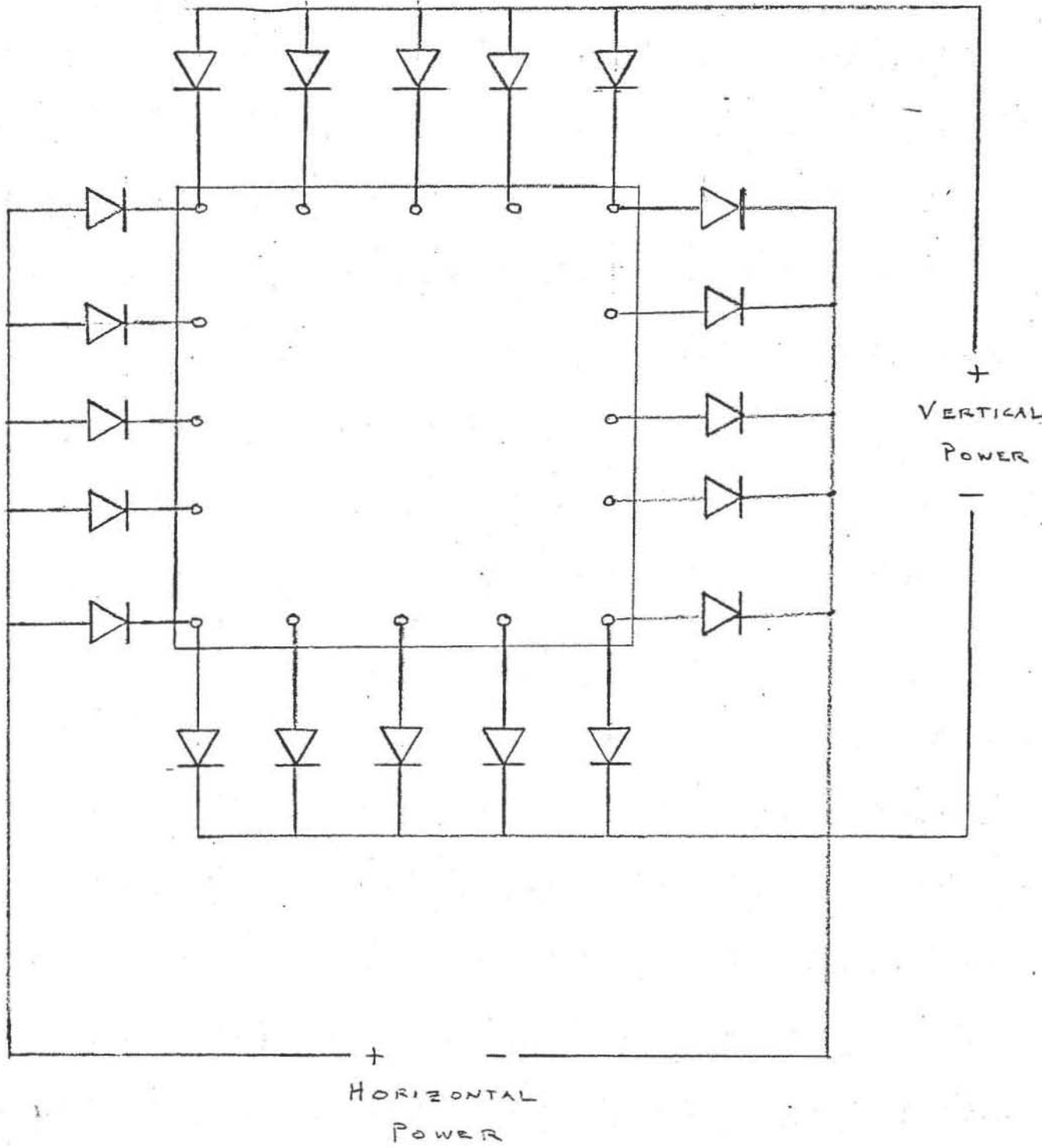
The stylus for the Rand Tablet has a small tip which is capacitively coupled to the wires to which it is closest. Within the stylus, a sensitive amplifier detects and amplifies these pulses and delivers them via coaxial cable to the logic box. The sequence of pulses coming from the stylus is a unique representation of the position of the stylus on the tablet. The sequence of pulses is put into a "Shift" register and then converted from the Gray-code to binary for delivery to the computer. In addition to detecting the position of the stylus, the pen has a small switch which detects whether or not its user is pressing down on the pen. The position of this switch is reported to the computer where it is commonly used to control the flow of logical "ink", i.e. to control whether or not the coordinates are stored in memory.

The Rand Tablet is generally placed on the desk in front of a vertically-mounted cathode ray tube display. It is customary for the

program to present a spot on the display in a position which corresponds to the position of the stylus on the Rand Tablet. The user of the tablet looks at the spot on the cathode ray tube but controls the motion of the spot by moving the stylus on the Rand Tablet. The hand-eye coordination required to write on one surface and look at another comes very naturally. Because the writing surface is separated from the point of observation, the hand used for writing does not cover the written material. Both the Rand Tablet and the version of the voltage gradient stylus built by Sylvania are transparent, and so they can be placed directly in front of the display if desired.

The Voltage-Gradient Stylus

Another position-detecting technique utilizes voltage gradients within a resistive plate. In its simplest configuration, a sheet of partly conductive material is used as the tablet surface. In successive time intervals, a potential is applied horizontally across this sheet and then vertically across the sheet. Diodes may be used in the connections to the edges of the sheet to prevent the vertical connections from distorting the horizontal field and vice-versa, as shown in the figure on the next page. The stylus, in actual contact with the conductive sheet, senses a potential which corresponds to its position on the sheet. By observing the potential during horizontal and vertical time periods, the associated electronics can determine the x and y coordinates of the pen. By observing whether or not there is a potential present at all, the electronics can determine whether the pen is in contact with the tablet surface.



The major difficulty in building a voltage-gradient stylus is obtaining a material suitable for the tablet surface. The material must be sufficiently tough to stand the wear of constant contact with the moving stylus. It must also have sufficiently uniform resistivity so that the potential measured is a linear function of the position. The problem is made more difficult by the need to have sufficiently high resistivity so that reasonable potentials can be developed across the tablet. Thus the conducting surface cannot be made of a copper plate which would be ideal in all other respects.

Sylvania has recently announced a tablet device similar to the voltage-gradient stylus. In the Sylvania device, the resistive sheet is a layer of stannous-oxide fused into glass plate and covered with another glass plate. Sylvania's engineers have shown that only seven contacts need be made to each edge of the plate in order to achieve one-percent precision. Moreover, they have worked out a technique for compensating for non-linearities in the plate by means of a few compensating resistors connected to these contacts. Sylvania is thus able to compensate individually for the difficulties in relatively poor conducting sheets to achieve the desired precision.

In Sylvania's device, the pen stylus does not actually contact the conductive sheet. The signals put in the plate are high-frequency alternating currents applied in such a way that the phase detected by the stylus varies for different positions on the sheet. Two different frequencies are used, one for horizontal sensing and one for vertical sensing. The phases of the two received signals, as filtered, are

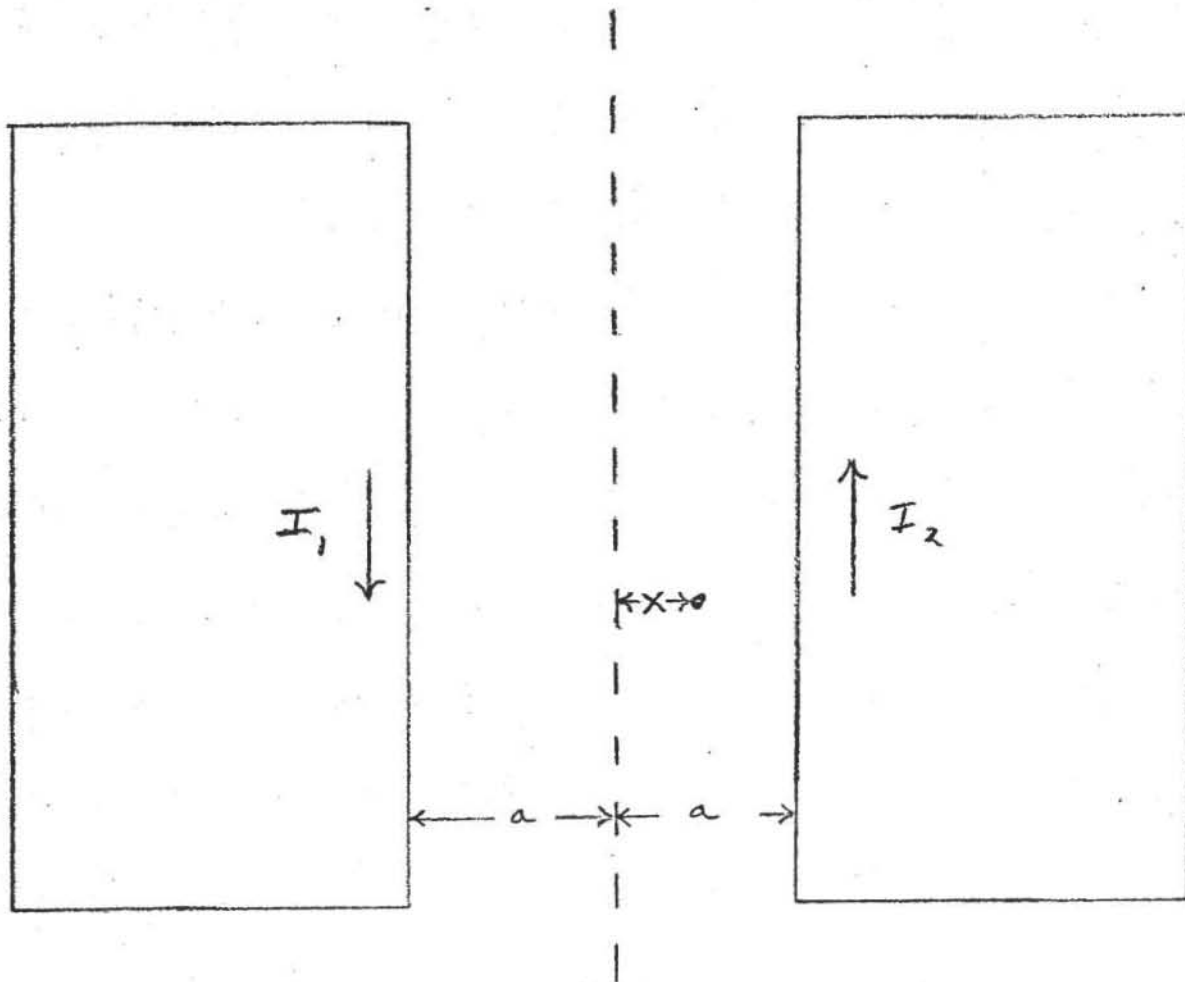
measured and correspond to the position of the stylus on the plate. Because high-frequency signals are used, there can be considerable separation between the stylus and the conducting surface. In fact, Sylvania's tablet works quite acceptably through a book. The magnitude of the received signal is measured and used to indicate height information to the computer. Three height signals are provided; one indicating that the pen is within about 1/32 inch of the surface, and one indicating that the stylus is actually being pressed down onto the surface. The final indication is given by a mechanical switch.

The Sylvania tablet is completely transparent. It is made of glass and the stannous-oxide coating is also transparent. It is in principle possible, therefore, to put the Sylvania tablet directly in front of the cathode ray tube. Rand tablets are translucent. Because the copper wires are relatively narrow and have relatively wide spacing, the Rand tablet will transmit about 50% of the light shining on it. An ideal display tablet combination would, I feel, be provided by projecting the cathode ray tube information from underneath a Sylvania or Rand tablet. The registration between display and tablet would need to be carefully controlled so that coordinates delivered from the tablet correspond, in detail, to the coordinates selected by the tablet.

Other Devices

Jack Raffel at the MIT Lincoln Laboratory has built a magnetic tablet which senses the relative strength of magnetic field coming from two

individual wires. The configuration of the magnetic tablet is as shown in the figure below.



The magnetic field in the region between the two loops varies roughly as $\frac{I}{r}$. The total magnetic field is therefore given by

$$M = \frac{I_1}{a+x} - \frac{I_2}{a-x}$$

In this case, the stylus pickup device is a small field-sensing loop. The current in one wire is gradually increased and the current in the other is gradually decreased until the field sensed by the loop changes sign.

$$I_1 = \frac{I_{\max}}{2} + Kt \qquad I_2 = \frac{I_{\max}}{2} - Kt \qquad -T < t < T$$

The time at which this happens is an indication of the position of the stylus on the tablet.

$$M = 0 = \frac{\frac{I_{\max}}{2} + Kt}{a+x} - \frac{\frac{I_{\max}}{2} - Kt}{a-x} \qquad x = \frac{4aK}{I_{\max}} t$$

The 3-D Wand

L. G. Roberts, then of the MIT Lincoln Laboratory, devised an ultrasonic device now known as the "Lincoln Wand". The Lincoln Wand senses positions in three dimensions, rather than in two dimensions as in the other devices mentioned. The Lincoln Wand is nothing but a hand-held microphone sensitive to high-frequency sound pulses. Four transducers mounted around the cathode ray tube display transmit pulses in turn at 8 millisecond intervals. The transit time of a pulse from the transmitter to the hand-held receiver is measured for each of the four paths, and from this information, the computer can deduce the position of the stylus.

The Lincoln Wand measures four distances rather than three in order to provide a check on the accuracy of the measurements. If the four distances measured are indicated by subscripts which indicate which quadrant of the display the corresponding transmitter is in, then

$$D_1^2 - D_2^2 + D_3^2 - D_4^2 = 0$$

as can easily be shown by geometric arguments. Moreover,

$$2ax = -D_1^2 + D_2^2 + D_3^2 - D_4^2 \quad \text{and}$$

$$2by = -D_1^2 - D_2^2 + D_3^2 + D_4^2 \quad \text{and}$$

$$4z^2 = 2(D_1^2 + D_2^2 + D_3^2 + D_4^2) - 4x^2 - 4y^2 - a^2 - b^2$$

where a and b are the x and y separation of the transducers.

As you can see, the computations of the coordinates of the stylus from the distance measurement information is not very difficult.

Comparator

A comparator is a device which examines the current position of the cathode ray tube beam and announces whenever that position is located within a certain region of interest, generally a small square. The comparator, then, computes the difference between the current position of the beam and the center of the small square in both x and y and produces a pulse whenever the magnitude of that difference is smaller than some tolerance in both x and y. Comparator devices should be designed as a part of display systems, except that historically they have nothing to do with stylus input at all.

$$2ax = -D_1^2 + D_2^2 + D_3^2 - D_4^2 \quad \text{and}$$

$$2by = -D_1^2 - D_2^2 + D_3^2 + D_4^2 \quad \text{and}$$

$$4z^2 = 2(D_1^2 + D_2^2 + D_3^2 + D_4^2) - 4x^2 - 4y^2 - a^2 - b^2$$

where a and b are the x and y separation of the transducers.

As you can see, the computations of the coordinates of the stylus from the distance measurement information is not very difficult.

(this goes mille 7. 10)

The comparator may take two forms. In the "center-size" form, the comparator has an "x" and a "y" register to store the coordinates for comparison and possibly also a register to store the tolerance, or at least an adjustment on the tolerance. In the "four edges" form, the coordinates of the top, bottom, left, and right edges of the sensitive area are stored in four registers. The registers of the comparator should be capable of being loaded under program control. In its most common use, the comparator registers will be loaded from the information derived from the stylus, but that should be a choice of the programmer and not a wired-in function. The programmer should be able to sensitize the comparator to whatever other values he chooses, such as for example, a position related to the position of the stylus but not exactly the position of the stylus. A comparator "hit" is treated logically in a fashion identical to that described for light pen hits.

POINTING DEVICES

The original stylus input devices were of the pointing type. They were called "light guns", so named because they looked like pistols and were aimed like a pistol at the cathode ray tube display. The light gun was developed so that operators could select particular targets of interest on the radar displays of early air defense systems. A version of the light gun, reduced to the size of a fountain pen, is now in common use. It is called the "light pen".

Both the light gun and the light pen contain a photo cell and a lens system. The lens system is so arranged that it focuses light from a small region of the CRT screen onto the photo cell. If the light pen or gun is aimed at the cathode ray tube and displayed information falls within the small field view of the light pen, light from the CRT will fall on the photo cell. Because the different parts of a picture on a cathode ray tube are displayed in time sequence, the time at which the photocell sees light corresponds to the particular object whose light has been sensed. The photo cell indicates that it has sensed the light by sending an electrical signal to the computer through a cable provided for that purpose. When it receives such a signal, commonly known as a "light pen hit", the computer can take appropriate action for the particular item in the picture then being displayed.

Logical Design Of Light Pens

There are two quite different kinds of hardware through which a light pen hit can be indicated to the computer. In pen systems with

only the simplest hardware, a light pen hit sets a simple flag which the computer can test if programmed to do so. In a more complex system, a light pen hit causes special hardware to start an interrupt procedure whether or not the main program was explicitly testing for light pen hits.

Each kind of light pen hardware can be programmed to provide the other function. If only the interrupt mechanism is provided in the hardware, an interrupt program which merely sets a bit in memory will provide the function of the light pen hit flag which the main program may test. The time cost of such a program is small because interrupts occur infrequently. It is far less convenient to provide the interrupt capability in software if only the flag is provided in the hardware, because the flag must be tested immediately after it posts each picture item on the CRT. Whenever a hit is detected, the main program should be forced to branch to an interrupt location. Although an interrupt generating program of this kind is easy to write and need occupy only little memory space, it will seriously decrease the speed of the display.

Because provision of interrupt through software is so costly, I consider a hardware essential if the light pen is intended to select objects in a picture. It is often useful, in addition, to have a light pen hit flag, and because the cost of a flag is small, there is no reason not to have both hardware capabilities. It should be possible under program control to mask off the interrupt mechanism if it is not needed. One can, of course, choose not to test a light pen hit flag.

It is interesting to note the relation between light pen hits in a display and unusual conditions which arise in other parts of a computer such as arithmetic overflow, memory violation, or I/O ready signals. For each such condition, two kinds of hardware may be provided: 1) a flag which can be tested by the program or 2) a device which starts an interrupt procedure. It is clear by now to all competent computer designers that interrupt procedures are desirable for input/output ready signals such as typewriter character ready, etc. It is also abundantly clear that interrupt procedures for handling arithmetic overflow are essential to efficient compiled code (though not all computers even today provide for interrupt on arithmetic overflow). What is not very clear, it seems, is that all these unusual condition testers could (and I maintain should) be handled in a perfectly uniform way. A single priority interrupt mechanism could scan all such conditions and initiate separate procedures appropriate to each.

I favor initiation of such procedures by the execution of the single instruction located at some particular place in memory. Please note that I did not say by transfer or branch (as is done in the SDS940) to a selected location, but rather by executing the instruction there. If the instruction there happens to be a "no op", the interrupt will effectively be ignored. If the instruction there is a subroutine branch instruction, the named interrupt procedure will have been begun. Automatic saving of the active machine registers (as is done in the PDP-1) is not strictly necessary; in a machine with many active registers it is less desirable than in a machine with only a few. In a machine with push down stack subroutines, interrupts may be permitted to interrupt others in a first-come, first-served basis if desired. Because it is usually important, however, to guarantee that an interrupt program will

finish its action within a certain limited time, the interrupt mechanism should include some mechanism for preventing low priority interrupts from disturbing a high priority process for synchronous I/O unit.

Because most displays fetch information from memory through data channels which are almost computers in their own right (See Chapter), a pen hit interrupt should probably affect only the display channel and not the main computer. Similarly, the light pen hit flag should be available to the display processor. Appropriate display channel programs can store information for later use by the main computer.

The most common use of light pen hits is to record the identity of the picture item that was indicated. In such an application, the light pen hit initiates an interrupt procedure which examines the interrupted display procedure to find out which item was being displayed. Typically, the interrupt procedure will record the memory address of the item which caused the hit. The interrupt procedure will deduce which item was seen from the content of the address registers which keep track of which item the main display procedure is to post next. If the information recorded is to accurately reflect the item seen, the light pen hardware must interrupt in such a way that consistent information is available. If a light pen hit from the first part of a line causes an interrupt before the address registers have been advanced, but a hit from the end of a line comes after the address registers have advanced, it will be difficult if not impossible to decide which line caused the hit.

The actual record made by the light pen interrupt procedure varies from application to application. In most cases the interrupt procedure

builds a small table of the items seen during a single frame. The content of the table will show if two or more items are being indicated as would happen if the light pen were aimed at the intersection of two lines. In some display systems, the interrupt procedure initiated by a light pen hit is a wired-in function. One display system (whose manufacturer shall remain unnamed) provides a wired-in interrupt procedure to record items seen by the light pen. Unfortunately, the wired-in procedure records only the first light pen hit of each frame thus making it virtually impossible to point to the intersection of lines or to do light pen tracking.

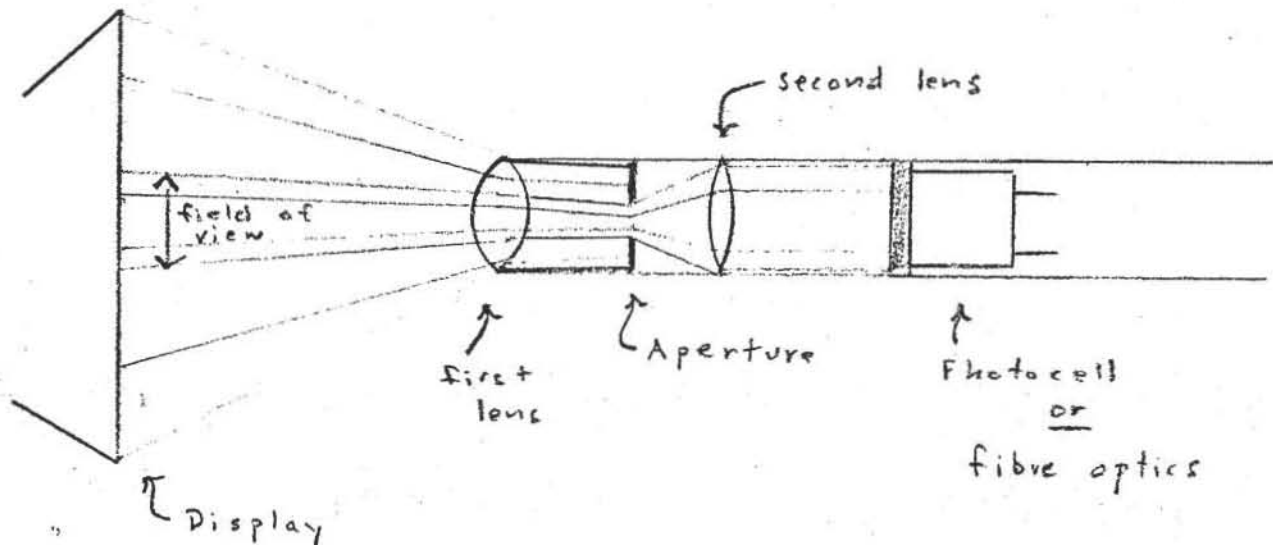
The record keeping job of the light pen interrupt program is made more complicated by the use of displays with subroutining capability. The address of the item being posted by such a display is not sufficient to identify it. If several symbol pictures, say of transistors, are posted on the picture by a single subroutine, an address within the subroutine will not identify which transistor was being displayed. What one really wants to record is both the subroutine involved and the route by which it was reached. For example, that the transistor is, in fact, the third one in the fourth flip flop. If the subroutine returns of the display system are kept neatly together in a stack, the light pen interrupt return can record a copy of the stack.

The information recorded by the light pen interrupt program must be double-buffered for use in any procedure which is not synchronous with the display. At the beginning of each frame, the interrupt program's hit table must be cleared. As the frame progresses, the hit table will grow. If an asynchronous procedure (in the main computer, for example)

asked whether a certain item had been seen, the answer might be "no" merely because the item had not yet been seen in the current frame even though it was seen in all previous frames. To avoid such a problem, the completed hit table must be copied into another buffer at the end of each display frame for use by the asynchronous process.

Physical Design Of Light Pens

The mechanical and optical design of a light pen is a more complex task than generally appreciated. A good light pen has a cylindrical field of view. That is, the area of the screen that it observes is circular and relatively constant in size, independent of how far from the screen the light pen is held. Achieving such a field of view is not an easy task. Many light pens are designed with a simple aperture and no lens system whatsoever. A more proper light pen lens system consists of two lenses, the first of which focuses the screen onto an aperture whose size and shape controls the size and shape of the field of view. The second lens focuses an image of the first lens onto the active area of the photo cell. Thus any light which passes through the first lens and through the aperture will be positioned on the photo cell according to the place it passed through the first lens.



The entire active area of the photo cell will be uniformly illuminated with an illumination dependent only on the proximity of the light source to the edge of the light pen field of view.

Some light pens are equipped with aiming lights which indicate the active area of the light pen by projection onto the screen. If the photo cell is mounted in the light pen housing, a coaxial cable is generally provided from the light pen to the housing of the cathode ray tube display to carry the electrical signal which indicates the presence of light. Some light pens replace the coaxial cable with a fibre-optic light pipe, and are thus able to use a larger and presumably more sensitive photo cell in the light pen electronic housing.

The most important electrical property of a light pen is its speed of response. The lines on the cathode ray tube are drawn relatively quickly, that is, in a few microseconds each, and so the light pen must respond in a fraction of a microsecond if it is to distinguish between successive parts of the picture displayed on the screen. Obtaining the required speed of response is the major difficulty in building a light pen. A significant obstacle is that the actual light output response to the phosphor is somewhat delayed from the control of the electron beam, and so even if the light pen were perfect, it might still not be good enough. As the speed of displays has increased, the usefulness of the light pen has correspondingly decreased to the point where it is now being replaced in research organizations by other types of stylus input devices.

The Light Cannon

Another interesting graphic input device is known as the "light cannon". In this device, a photomultiplier tube is placed in front of the cathode ray tube in such a way that it can sense light from anywhere on the display face. If an opaque object is placed between the cathode ray tube and the light cannon, it will shield certain regions of the cathode ray tube from observation. Each point displayed on the cathode ray tube face will be sensed by the photo-multiplier only if it is not shielded. An opaque object can be used to point out particular items on the cathode ray tube by shielding them from observation of the light cannon. The light cannon can also be used to sense the shape of irregular objects placed in front of the cathode ray tube's screen, or with special adaptation, to scan photographic material.

APPENDIX I

" T R A N S F O R M A T I O N S A N D M A T R I C E S "

STEVEN A. COONS

Presented as part of the 1967 Summer
Conference on Computer Graphics for
Designers at the University of Michigan

TRANSFORMATIONS AND MATRICES

We represent the coordinates of a point in two dimensions by the matrix $\begin{bmatrix} x & y \end{bmatrix}$. The elements of this matrix are independent, and the pair of numbers constitute a matrix quantity.

Now consider the matrix product of such a coordinate matrix and a 2×2 matrix:

$$\begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = [(ax+cy) \ (bx+dy)] = [x' \ y'] .$$

The result of the matrix multiplication consists again of two numbers, $(ax+cy)$ and $(bx+dy)$, and we can investigate the implications of assuming that these two numbers are new coordinates x' and y' .

In passing, we should remark that the matrix multiplication consists of multiplying a row matrix $\begin{bmatrix} x & y \end{bmatrix}$ by the column matrix $\begin{bmatrix} a \\ c \end{bmatrix}$ to yield $(ax+cy)$ and by the column matrix $\begin{bmatrix} b \\ d \end{bmatrix}$ to yield $(bx+dy)$. We can thus think of the square matrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ as consisting of two separate column matrices. Incidentally, row matrices like $\begin{bmatrix} x & y \end{bmatrix}$ or $\begin{bmatrix} x & y & z & w \end{bmatrix}$ are commonly called vectors, as are column matrices like $\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$,

and the rule for formation of their product is

$$\begin{bmatrix} x & y & z & w \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = ax + by + cz + dw .$$

The product of two more general matrices follows from this; we could have the product

$$\begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \end{bmatrix} \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \\ c_1 & c_2 \end{bmatrix} = \begin{bmatrix} (a_1x_1 + b_1y_1 + c_1z_1)(a_2x_1 + b_2y_1 + c_2z_1) \\ (a_1x_2 + b_1y_2 + c_1z_2)(a_2x_2 + b_2y_2 + c_2z_2) \end{bmatrix}$$

and we can think of the product as consisting of the various products of the two row vectors in the first matrix and the two column vectors in the second matrix.

To return to the matrix product of the vector matrix $[x \ y]$ and the 2×2 matrix, we can investigate some simple special cases, and see what the geometric interpretation is.

Take

$$[x \ y] \begin{bmatrix} a & 0 \\ 0 & 1 \end{bmatrix} = [ax \ y] = [x' \ y']$$

The new coordinates of a point $[x \ y]$ are similar to the old coordinates, but with a scale change in x . This means that the act of multiplying by the matrix has the effect of stretching the original figure, whatever it may be, in the x direction.

Now take

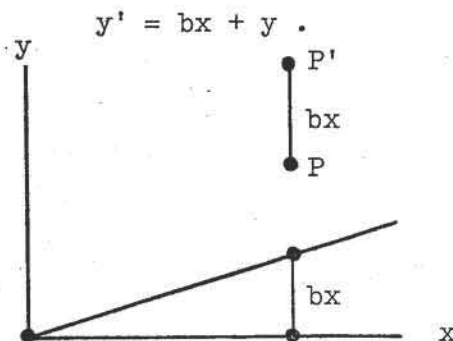
$$[x \ y] \begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix} = [ax \ dy] = [x' \ y'] .$$

This represents a scale change in both x and y . The geometry of the original figure has experienced a stretching in the x direction and a simultaneous stretching in the y direction. Of course, if either a or d is fractional, less than 1, the result is a compression of the original figure.

Now consider

$$[x \ y] \begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix} = [x \ (bx + y)] = [x' \ y'] .$$

Here the old x and the new x' coordinates are the same, but the coordinate y' is given by the linear equation



The quantity bx is the amount by which the old y coordinate is increased to give the new y' coordinate. The x coordinate is unchanged.

In particular, suppose we take a unit square in the original coordinate system and apply this transformation to it. The four corners of the square are given by the four vectors arranged in a matrix.

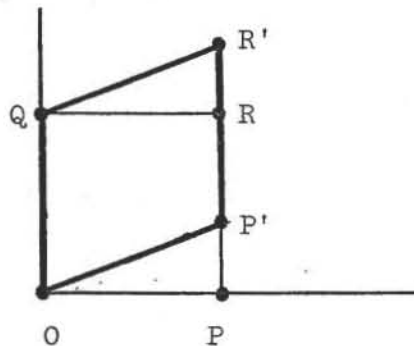
We can identify these points as follows:

$\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$	The origin of coordinates, O
$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$	The unit point on the x axis, P
$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$	The unit point on the y axis, Q
$\begin{bmatrix} 1 & 1 \end{bmatrix}$	The fourth corner of the square, R .

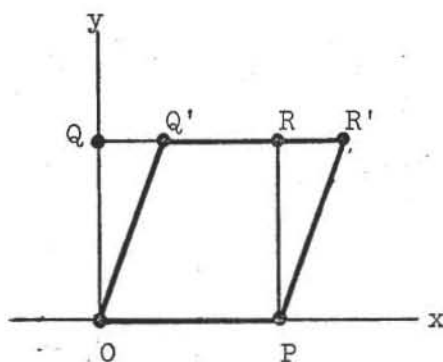
Then we form the matrix product:

$$\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & b \\ 0 & 1 \\ 1 & b+1 \end{bmatrix} \begin{matrix} O' \\ P' \\ Q' \\ R' \end{matrix}$$

We plot the old and new points:



Note that the points O and Q are unchanged, but points P and R are transformed into new points P' and R' . The square has been transformed into a parallelogram, and it is customary to refer to this transformation as a "shear"; the figure has been "sheared" in the y direction by an amount b for each unit of x . Similarly the matrix $\begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix}$ yields a shear transformation in the x direction, and a unit square will transform into a parallelogram:



where in this case O and P remain fixed, but Q and R transform into new points:

$$\begin{matrix} Q \\ R \end{matrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix} = \begin{bmatrix} c & 1 \\ 1+c & 1 \end{bmatrix} \begin{matrix} Q' \\ R' \end{matrix}$$

Let us now combine these two transformations. We shall elect to perform the y shearing transformation first, and then we shall perform the x shearing transformation on the result. For a general point, this is accomplished by the matrix multiplication

$$[x \ y] \begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix} = [x' \ y']$$

followed by the matrix multiplication

$$[x' \ y'] \begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix} = [x'' \ y''] .$$

Then we can write the combined operation

$$[x \ y] \begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix} = [x'' \ y''] .$$

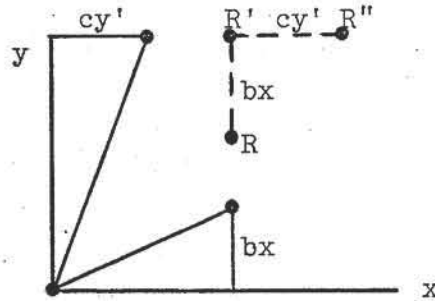
But we can evaluate the matrix product of the two separate shear transformations,

$$\begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix} = \begin{bmatrix} 1+bc & b \\ c & 1 \end{bmatrix}$$

Now, subject to this transformation, the unit square transforms as follows:

$$\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1+bc & b \\ c & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1+bc & b \\ c & 1 \\ 1+bc+c & b+1 \end{bmatrix}$$

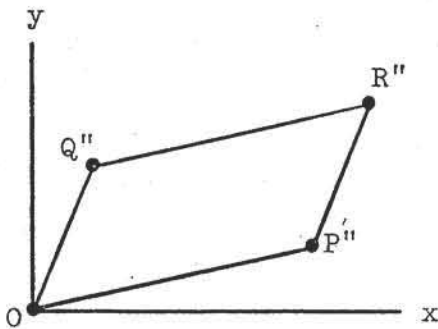
Consider only the point R_1 and the two transformations applied to it:



The first transformation moves R vertically by the amount b . The second transformation moves R' horizontally by an amount that is proportional to its y' position. This horizontal motion is cy' , but $y' = 1+b$, so the horizontal motion of R' is $c+bc$. Its final position is

$$R'' = [1+bc+c \quad 1+b] .$$

We observe that the relative positions of the four points after the transformation are given by the difference of their vectors, as follows:



$$[OQ''] = [Q''] - [O]$$

$$[P''R''] = [R''] - [P'']$$

$$[OP''] = [P''] - [O]$$

$$[Q''R''] = [R''] - [Q'']$$

where the symbols in brackets stand for the matrices of the corresponding coordinates.

$$[OQ''] = [c \ 1] - [0 \ 0] = [c \ 1]$$

$$[P'' \ R''] = [(1+bc+c)(b+1)] - [(1+bc) \ b] = [c \ 1] .$$

Hence $[OQ''] = [P'' R'']$

Similarly,

$$[OP''] = [(1+bc) b] - [0 0] = [(1+bc) b]$$

$$[Q'' R''] = [(1+bc+c) (b+1)] - [c 1] = [(1+bc) b] .$$

Hence $[OP''] = [Q'' R'']$.

These two expressions indicate that the result of the transformation is a parallelogram, but in a general position in the coordinate system.

Now consider the points P and Q before the transformation, given by $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and the points P'' and Q'' after the transformation,

given by $\begin{bmatrix} 1+bc & b \\ c & 1 \end{bmatrix}$. This last matrix is not only the matrix describing

the final position of the two unit points P and Q , but it is also the matrix of the transformation that carries these points into this final position. We can see that this is invariably so, for any transformation,

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} .$$

Here the new P' = [a b] and the new Q' = [c d] .

If we combine the two shearing transformations with two scale changes, we obtain

$$\begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix} \begin{bmatrix} a & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & d \end{bmatrix} = \begin{bmatrix} a+abc & bd \\ ac & d \end{bmatrix}$$

We can proceed to show that this represents the most general possible 2 x 2 transformation matrix, as follows: Select any four numbers, A B C D , arrange them in a matrix, and form the matrix equation:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} a+abc & bd \\ ac & d \end{bmatrix} .$$

Then, equating corresponding elements,

$$\begin{aligned} d &= D \\ bd &= B \text{ whence } b = \frac{B}{D} . \end{aligned}$$

$$C = ac, \text{ and } A = a+abc = a + b(ac) = a + \frac{BC}{D} .$$

Then $a = A - \frac{BC}{D}$. Finally, $c = \frac{C}{a} = \frac{C}{A - \frac{BC}{D}}$. Thus the four numbers $a b c d$ may be suitably chosen so as to make the resulting combined transformation equal to any transformation whatever.

Ordinarily we do not build up transformations out of their constituent elementary transformations. Instead, we determine the initial and final positions of certain important points, and deduce from this what the appropriate transformation must be to yield this result.

In order to be able to do this, we first turn our attention to the transformation

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \text{ that carries the points } \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

into new positions, described by the transformation matrix itself, and another transformation that undoes the transformation and restores the points to their original positions. Let this second transformation be

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

and we want to choose its elements so that

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

We evaluate the indicated matrix product, and obtain

$$\begin{bmatrix} aA + bC & aB + bD \\ cA + dC & cB + dD \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

There are in effect two equations involving A and C , and two other equations involving B and D . When we solve these two systems for the four unknowns, we obtain

$$A = \frac{d}{ad-bc} \quad B = \frac{-b}{ad-bc} \quad C = \frac{-c}{ad-bc} \quad D = \frac{a}{ad-bc}.$$

The quantity $\frac{1}{ad-bc}$ is common to all of these, and we may write the definitive matrix equation

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

The quantity $ad-bc$ is precisely the determinant of the matrix. Of course if $ad-bc = 0$, we obtain no meaningful result.

We can check the validity of this new matrix by sets of multiplication:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \begin{bmatrix} ad-bc & 0 \\ 0 & ad-bc \end{bmatrix}$$

and the resulting matrix is $(ad-bc)\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. The matrix $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$ as just evaluated is called the inverse of the matrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$. In general, a matrix will have an inverse provided its determinant does not vanish.

It turns out that higher order matrices can be inverted by an entirely analogous procedure. We form, for each element of the original matrix, its cofactor, which is the determinant of the matrix obtained by crossing out the row and column in which the element appears. Thus the cofactor of the element b in the

matrix $\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$ is the determinant of the matrix $\begin{bmatrix} d & f \\ g & i \end{bmatrix}$. We

write this number in place of b , and we multiply it by $(-1)^{C+R}$ where $C+R$ is the sum of the number of the column and the number of the row in which the element appears. In the case of b , $C = 2$, $R = 1$, $(-1)^{C+R} = (-1)^3 = -1$. Hence the cofactor of b is the determinant of the sub-matrix with a minus sign. We thus obtain a cofactor matrix. Then the inverse desired is the transpose of this cofactor matrix (in which we simply interchange rows and columns) divided by the determinant of the original matrix. In the simple case of the

2×2 matrix, the cofactor matrix of $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ is $\begin{bmatrix} d & -c \\ -b & a \end{bmatrix}$,

and its transpose is $\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$, the desired inverse.

Now suppose we wish to find a transformation T that carries points $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ into $\begin{bmatrix} a' & b' \\ c' & d' \end{bmatrix}$. The transformation involves the matrix product

$\begin{bmatrix} a & b \\ c & d \end{bmatrix} T = \begin{bmatrix} a' & b' \\ c' & d' \end{bmatrix}$, where we are using the symbol T to replace an unknown 2×2 transformation matrix. Suppose we were to pre-multiply both sides of this equation by the inverse of the matrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$.

We use a superscript -1 to indicate the inverse of a matrix, and write

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} \begin{bmatrix} a & b \\ c & d \end{bmatrix}^T = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} \begin{bmatrix} a' & b' \\ c' & d' \end{bmatrix}$$

But
$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \frac{1}{\Delta} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \frac{1}{\Delta} \begin{bmatrix} \Delta & 0 \\ 0 & \Delta \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

(we have used the symbol Δ to stand for the determinant of the matrix).

T is a matrix, so that

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^T = T = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} \begin{bmatrix} a' & b' \\ c' & d' \end{bmatrix}.$$

$$T = \frac{1}{\Delta} \begin{bmatrix} d & -d \\ -c & a \end{bmatrix} \begin{bmatrix} a' & b' \\ c' & d' \end{bmatrix}.$$

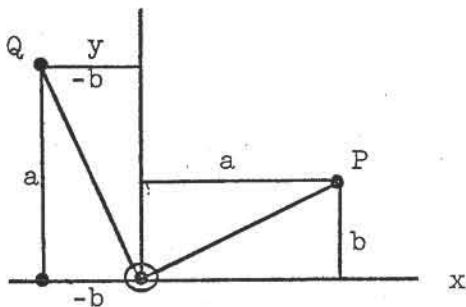
The result, if multiplied out, is a 2×2 transformation matrix, as desired.

ROTATION

A transformation of considerable importance is represented by the matrix

$$\begin{bmatrix} a & b \\ -b & a \end{bmatrix}.$$

We can gain an insight into its geometric interpretation if we plot the two points represented:



$$P = [a \ b], \quad Q = [-b \ a].$$

In the graph, the two vectors OP and OQ are obviously perpendicular to one another, and the points P and Q are equidistant from O . If, furthermore, this distance is $a^2 + b^2 = 1$, then the

transformation represents a pure rotation of the unit points on the two axes. In this case, a is the cosine of the angle of rotation, and b is the sine of that angle. The transformation matrix becomes

$$\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

and we have the familiar rotation formulas:

$$x' = x \cos\theta - y \sin\theta$$

$$y' = x \sin\theta + y \cos\theta ,$$

which we can obtain by multiplication of the vector $[x \ y]$ by the matrix.

The inverse of the matrix is, by substituting in our previous result,

$$\begin{bmatrix} a & b \\ -b & a \end{bmatrix}^{-1} = \frac{1}{\Delta} \begin{bmatrix} a & -b \\ b & a \end{bmatrix}$$

where $\Delta = a^2 + b^2$. But observe that the matrix on the right of the equation is simply the transpose of the matrix on the left, since it is this matrix with rows and columns interchanged. If $\Delta = 1$, we have the equation

$$\begin{bmatrix} a & b \\ -b & a \end{bmatrix}^{-1} = \begin{bmatrix} a & b \\ -b & a \end{bmatrix}^T$$

where the T superscript means "transpose." This indicates that it is a simple matter to form the inverse of a rotation transformation; all we need do is write its transpose.

TRANSLATION

~~TRANSACTION~~

We have thus far investigated transformations which change points in the plane, but leave one point unchanged. This point is the origin of coordinates. We shall now investigate the pure translation of points in the plane. Consider the following matrix product:

$$[x \ y \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a & b & 1 \end{bmatrix} = [(x+a)(y+b)1] = [x' \ y' \ 1] .$$

By the algebraic artifice of introducing the number 1 into the point coordinate vector, and by expanding the transformation matrix from a 2 x 2 into a 3 x 3 matrix, we are able to slide the original figure into a new position in which the origin of coordinates is also moved:

$$[0 \ 0 \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a & b & 1 \end{bmatrix} = [a \ b \ 1] .$$

What would be the effect of performing first a general transformation that left the origin unchanged, and then subsequently translating the entire resulting figure? We can try this experiment out as follows.

$$[x \ y \ 1] \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 1 \end{bmatrix} = [(ax+cy) \ (bx+dy) \ 1] = [x' \ y' \ 1] .$$

Then,

$$[x' \ y' \ 1] \begin{bmatrix} 1 & 0 & 0 \\ e & f & 1 \\ 0 & 0 & 1 \end{bmatrix} = [(x'+e) \ (y'+f) \ 1] = [x'' \ y'' \ 1] .$$

The combined transformation is given by the matrix product of the separate transformations:

$$\begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ e & f & 1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{bmatrix} .$$

Incidentally, if we perform the translation first, followed by the general origin-preserving transformation, we get a different result:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ e & f & 1 \end{bmatrix} \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ (ea+fc) & (eb+fd) & 1 \end{bmatrix}$$

The resulting matrix shows that the origin $[0 \ 0 \ 1]$ transforms into the point $[(ea+fc)(eb+fd) \ 1]$.

THREE DIMENSIONAL TRANSFORMATIONS

The matrix

$$\begin{bmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{bmatrix}$$

which represents a general two dimensional

transformation together with a translation is obviously a special case of the three dimensional transformation

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = T .$$

in general, $[x \ y \ z]T = [x' \ y' \ z']$.

As with the two dimensional case, we select unit points on the three axes:

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and when this matrix of coordinates is multiplied by}$$

T, we obtain the transformed coordinates of the three unit points; moreover, the transformation matrix itself consists of the three vectors of coordinates of the transformed points, P', Q' and R'.

Again, as in the case of the origin of coordinates in two dimensional, the origin remains fixed, for

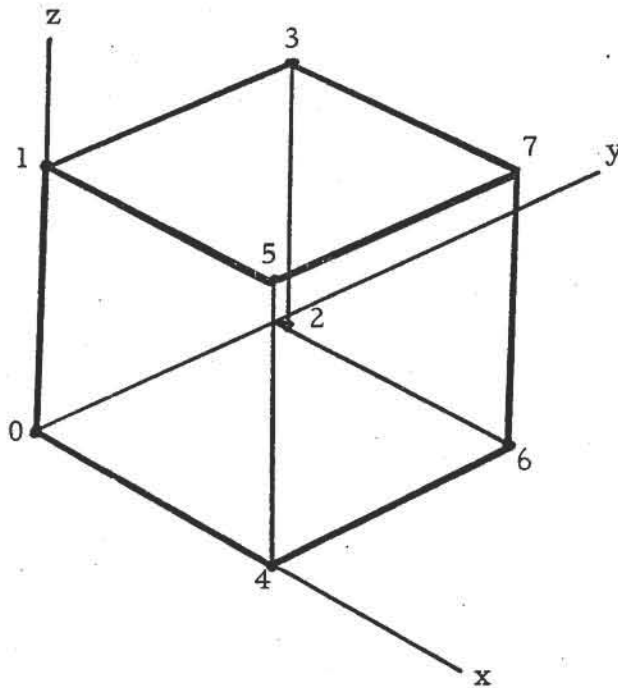
$$[0 \ 0 \ 0]T = [0 \ 0 \ 0].$$

We can now attach a more illuminating meaning to the two dimensional translation transformation. The vector $[x \ y \ 1]$ represents points on the plane $z = 1$. The translation transformation keeps z fixed, but allows x and y to change in this plane. The origin of x and y coordinates is given by the vector $[0 \ 0 \ 1]$, which after the transformation becomes $[e \ f \ 1]$. But the origin of the entire three dimensional system is $[0 \ 0 \ 0]$ and this origin remains fixed.

We can show that the unit cube transforms into a parallelepiped in three dimensional space. The transformation is as follows:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ g & h & i \\ d & e & g \\ (d+g) & (e+h) & (f+i) \\ a & b & c \\ (a+g) & (b+h) & (c+i) \\ (a+d) & (b+e) & (c+f) \\ (a+d+g) & (b+e+h) & (c+f+i) \end{bmatrix}$$

The points on the corners of the cube in the matrix on the left have been chosen in a particular order; it will be observed that they have been arranged in numerical sequence, when we count these corners base two.



This sequence of points is shown in the figure, where the numbers at the corners are base 10, equivalent of the base two numbers.

By detailed examination, we can assure ourselves that after the transformation the edge vectors are equal in such a way as to satisfy the vector equations of the following scheme. In these equations, the numbers in

brackets stand for the vectors of the numbered points. With this notation in mind we write:

$$[1 - 0] = [3 - 2] = [7 - 6] = [5 - 4]$$

$$[2 - 0] = [3 - 1] = [7 - 5] = [6 - 4]$$

$$[4 - 0] = [5 - 1] = [7 - 3] = [6 - 2] .$$

For example,

$$\begin{aligned} [5 - 1] &= [(a + g)(b + h)(c + i) - [g h i]] \\ &= [a b c] = [4 - 0] . \end{aligned}$$

This set of vector equations insures that the faces of the figure after the transformation are all parallelograms, and hence that the original cube transforms into a parallelepiped.

The transformation matrix is entirely defined by three points, whose coordinates are known both before and after the transformation. We can write

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \cdot T = \begin{bmatrix} a' & b' & c' \\ d' & e' & f' \\ g' & h' & i' \end{bmatrix}$$

where the primes indicate the transformed coordinates.

Hence

$$T = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^{-1} \begin{bmatrix} a' & b' & c' \\ d' & e' & f' \\ g' & h' & i' \end{bmatrix} .$$

This last equation requires the calculation of the inverse of a 3×3 matrix.

If we elect to do this by cofactors, we obtain first the new cofactor matrix

$$C = \begin{bmatrix} \begin{vmatrix} e & f \\ h & i \end{vmatrix} & -\begin{vmatrix} d & f \\ g & i \end{vmatrix} & \begin{vmatrix} d & e \\ g & h \end{vmatrix} \\ -\begin{vmatrix} b & c \\ h & i \end{vmatrix} & \begin{vmatrix} a & c \\ g & i \end{vmatrix} & -\begin{vmatrix} a & b \\ g & h \end{vmatrix} \\ \begin{vmatrix} b & c \\ e & f \end{vmatrix} & -\begin{vmatrix} a & c \\ d & f \end{vmatrix} & \begin{vmatrix} a & b \\ d & e \end{vmatrix} \end{bmatrix}$$

The inverse is then the transpose of this matrix, divided by the determinant of the complete original matrix:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^{-1} = \frac{1}{\Delta} C^T$$

For 3×3 matrices, the method of cofactors is marginally efficient, but inversion of higher order matrices becomes increasingly involved, since it requires calculation of a great many determinants of high order. For this reason, many numerical schemes are in existence designed for both hand calculation and computer evaluation. Usually such schemes depend upon relaxation methods, in which a number of simple iterations cause an original approximate solution to converge toward a more exact solution.

PROJECTIVE TRANSFORMATIONS

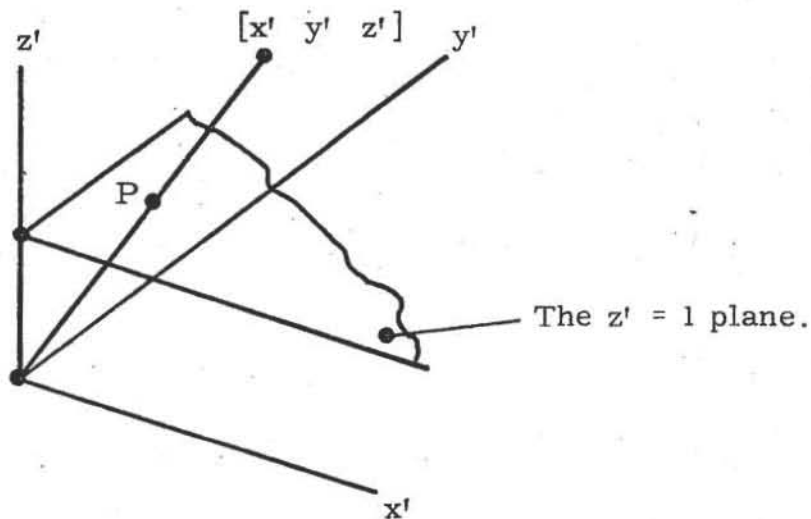
Consider the matrix transformation

$$[x \ y \ 1] \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = [x' \ y' \ z'].$$

The third coordinate of the transformed point, z' , is given by $z' = cx + fy + i$.

Now consider the figure obtained by dividing the vector $[x' \ y' \ z']$ by z' . The result is

$$\begin{bmatrix} \frac{x'}{z'} & \frac{y'}{z'} & 1 \end{bmatrix} = P.$$



The point $[x' \ y' \ z']$ has, by this division, been "projected" into point P in the plane $z' = 1$ by a projection ray through the origin of coordinates.

The coordinates of P in this $z' = 1$ plane are given by $\frac{x'}{z'} \ \frac{y'}{z'} \ 1$, and the resulting figure, of which this is a typical point, is two dimensional.

We often refer to the coordinates $[x' \ y' \ z']$ as the homogeneous coordinates of a point in two dimensions. They are also the ordinary coordinates of a point in three dimensions.

Consider the equation

$$Ax + By + C = 0.$$

This is the inhomogeneous equation of a line in two dimensions. In vector form it is

$$[x \ y \ 1] \begin{bmatrix} A \\ B \\ C \end{bmatrix} = 0.$$

If C is not zero, this can be rewritten:

$$[x \ y \ 1] \begin{bmatrix} \frac{A}{C} \\ \frac{B}{C} \\ 1 \end{bmatrix} = 0,$$

and this is as meaningful as the first form. But we now wonder whether by the artifice of making the equation homogeneous, we might not be able to obtain some useful generality. So we write

$$[x \ y \ w] \begin{bmatrix} A \\ B \\ C \end{bmatrix} = 0.$$

We have now made the point-coordinate vector homogeneous by introducing the third coordinate w . The ordinary coordinates of a point are always obtainable, because

$$X = \frac{x}{w} \quad \text{and} \quad Y = \frac{y}{w}.$$

But we have an added advantage, because if w is zero, the point $[x \ y \ 0]$ is a point at infinity.

The two factors of the matrix product have special significance. The vector $[x \ y \ w]$ is a point-coordinate vector; for $A \ B \ C$ fixed, all number triplets that satisfy $[x \ y \ w] \begin{bmatrix} A \\ B \\ C \end{bmatrix} = 0$ are coordinates of points on a fixed

line. Conversely, if $[x \ y \ w]$ are three fixed numbers, then all number triplets $A \ B \ C$ that satisfy the equation represent lines through the fixed point. For this reason, we refer to the transpose of the vector $[A \ B \ C]$ as a line vector, consisting of the line coordinates A , B , and C . When $C = 0$, the line passes through the origin and we have

$$[x \ y \ w] \begin{bmatrix} A \\ B \\ 0 \end{bmatrix} = Ax + By = 0 \text{ as we might expect.}$$

We also have, for $w = 0$

$$[x \ y \ 0] \begin{bmatrix} A \\ B \\ C \end{bmatrix} = Ax + By = 0.$$

Since the ordinary coordinates of a point are given by $X = \frac{x}{w}$ and $Y = \frac{y}{w}$, we shall modify the notation slightly in what follows. We shall write, for point vectors

$$[wx \ wy \ w] \text{ instead of } [x \ y \ w]$$

In this way, we shall be able to keep track of the ordinary coordinates of a point; we shall consider wx and wy as biliteral symbols throughout all calculations, and shall perform the operations $x = \frac{wx}{w}$ and $y = \frac{wy}{w}$ only at the very end. If, as will sometimes happen, $w = 0$, then we shall not attempt to perform this division, but will accept the numbers wx and wy as our result, and we shall know in this case that the point in question is at infinity.

Now consider the transformation

$$[wx \ wy \ w] \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = [w'x' \ w'y' \ w'] .$$

This may be thought of either as a three-dimensional origin-fixed transformation, in which the ordinary coordinates of a point are wx , wy , and w ; or on the other hand it can be thought of as a two-dimensional transformation in homogeneous coordinates. In this latter case, the transformation carries one plane figure into another.

In homogeneous coordinates, the matrix of point vectors

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

represents the point at infinity on the x axis, the point at infinity on the y axis, and the origin of coordinates in the plane $w = 1$. After the transformation, these three points become

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

and if c, f, i are not zero, the points have ordinary coordinates

$$\begin{bmatrix} \frac{a}{c} & \frac{b}{c} & 1 \\ \frac{d}{f} & \frac{e}{f} & 1 \\ \frac{g}{i} & \frac{h}{i} & 1 \end{bmatrix}$$

in the plane $w' = 1$.

Let us now consider the transformation of three points into three other points:

$$\begin{bmatrix} x'_1 & y'_1 & 1 \\ x'_2 & y'_2 & 1 \\ x'_3 & y'_3 & 1 \end{bmatrix} \begin{bmatrix} T \end{bmatrix} = \begin{bmatrix} w_1 x_1 & w_1 y_1 & w_1 \\ w_2 x_2 & w_2 y_2 & w_2 \\ w_3 x_3 & w_3 y_3 & w_3 \end{bmatrix}$$

$$\begin{bmatrix} T \end{bmatrix} = \begin{bmatrix} x'_1 & y'_1 & 1 \\ x'_2 & y'_2 & 1 \\ x'_3 & y'_3 & 1 \end{bmatrix}^{-1} \begin{bmatrix} w_1 x_1 & w_1 y_1 & w_1 \\ w_2 x_2 & w_2 y_2 & w_2 \\ w_3 x_3 & w_3 y_3 & w_3 \end{bmatrix}$$

We can invert the first matrix. The second matrix consists of ordinary coordinates

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix}$$

and three homogeneous coordinates w_1 , w_2 and w_3 . These are unknown, as yet, even though the desired positions of these three points is specified. We can choose any numbers we please for w_1 , w_2 , w_3 , including zero, if we wish, and then the transformation T will be defined. For some such choice of the w 's, a fourth point $[x'_4 \ y'_4 \ 1]$ will transform $[w_4 x_4 \ w_4 y_4 \ w_4]$, in a unique way. This suggests that we deliberately choose such a fourth point, and cause it to transform into some desired position. This will give us information about the quantities w_1 , w_2 , and w_3 .

A numerical example will be illuminating here. Let us transform the three points

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

into themselves. These are, respectively, the origin of $x'y'$ coordinates, the unit point on the y' axis, and the unit point on the x' axis. We have:

$$T = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 0 & w_1 \\ 0 & w_2 & w_2 \\ w_3 & 0 & w_3 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & w_1 \\ 0 & w_2 & w_2 \\ w_3 & 0 & w_3 \end{bmatrix}$$

The transformation T is only partly defined.

For the fourth point, take $[x'_4 \ y'_4 \ 1] = [1 \ 1 \ 1]$ and let it transform into the point with ordinary coordinates $[x_4 \ y_4 \ 1] = [2 \ 2 \ 1]$.

The homogeneous coordinates of the fourth point are

$$[w_4 x_4 \ w_4 y_4 \ w_4] = w_4 [2 \ 2 \ 1].$$

We shall drop the subscript from w_4 . It will turn out that we could have set $w_4 = 1$ at this stage, but we shall retain it for the time being.

We now have the matrix equation of the transformation

$$\begin{aligned} w[2 \ 2 \ 1] &= [1 \ 1 \ 1]T = [1 \ 1 \ 1] \begin{bmatrix} -1 & 0 & 1 \\ -1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & w_1 \\ 0 & w_2 & w_2 \\ w_3 & 0 & w_3 \end{bmatrix} \\ &= [-1 \ 1 \ 1] \begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_2 & 0 \\ 0 & 0 & w_3 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \end{aligned}$$

$$w[2 \ 2 \ 1] \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}^{-1} = [-1 \ 1 \ 1] \begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_2 & 0 \\ 0 & 0 & w_3 \end{bmatrix}$$

On the left, we have

$$w[2 \ 2 \ 1] \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}^{-1} = w[2 \ 2 \ 1] \begin{bmatrix} -1 & 0 & 1 \\ -1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} = w[-3 \ 2 \ 2].$$

On the right, we have

$$[-1 \ 1 \ 1] \begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_2 & 0 \\ 0 & 0 & w_3 \end{bmatrix} = [-w_1 \ w_2 \ w_3].$$

This leads to the vector equation

$$w[-3 \ 2 \ 2] = [-w_1 \ w_2 \ w_3]$$

from which

$$w_1 = 3w$$

$$w_2 = 2w$$

$$w_3 = 2w.$$

The Transformation T is now given by

$$T = w \begin{bmatrix} -1 & 0 & 1 \\ -1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 3 \\ 0 & 2 & 2 \\ 2 & 0 & 2 \end{bmatrix} = w \begin{bmatrix} 2 & 0 & -1 \\ 0 & 2 & -1 \\ 0 & 0 & 3 \end{bmatrix}$$

There is an arbitrary constant w involved, which we shall continue to carry along. We check the transformation, to see whether it does indeed transform the four points into their desired positions:

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} w \begin{bmatrix} 2 & 0 & -1 \\ 0 & 2 & -1 \\ 0 & 0 & 3 \end{bmatrix} = w \begin{bmatrix} 0 & 0 & 3 \\ 0 & 2 & 2 \\ 2 & 0 & 2 \\ 2 & 2 & 1 \end{bmatrix}$$

The ordinary coordinates of the four points are seen to be

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 2 & 2 & 1 \end{bmatrix},$$

a result we obtain by dividing through each row of the matrix by the corresponding homogeneous coordinate in the last column. The common multiplier w has had no effect on the result, and we now see that we could have set it equal to 1 at the outset.

Now let us see what this transformation does to lines in the plane.

The equation of a line before the transformation is

$$[w'x' \quad w'y' \quad w'] \begin{bmatrix} A' \\ B' \\ C' \end{bmatrix} = 0.$$

After the transformation, points are transformed according to

$$[w'x' \quad w'y' \quad w']T = [wx \quad wy \quad w].$$

We can preserve the linear equality if we write

$$[w'x' \quad w'y' \quad w']T T^{-1} \begin{bmatrix} A' \\ B' \\ C' \end{bmatrix} = 0$$

because

$$TT^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

the identity matrix. Introducing this matrix between the two vectors does not destroy the validity of the equation. But then

$$T^{-1} \begin{bmatrix} A' \\ B' \\ C' \end{bmatrix} = \begin{bmatrix} A \\ B \\ C \end{bmatrix} ,$$

the transformed line vector. After the transformation we have a new valid linear equation

$$[wx \quad wy \quad w] \begin{bmatrix} A \\ B \\ C \end{bmatrix} = 0 .$$

Now

$$T^{-1} = \begin{bmatrix} 2 & 0 & -1 \\ 0 & 2 & -1 \\ 0 & 0 & 3 \end{bmatrix}^{-1} = \frac{1}{6} \begin{bmatrix} 3 & 0 & 1 \\ 0 & 3 & 1 \\ 0 & 0 & 2 \end{bmatrix} .$$

Let us apply the transformation to the line at infinity of the original system, and find the equation of the transformation of this line in the new system.

The equation of the line at infinity is

$$[w'x' \quad w'y' \quad w'] \begin{bmatrix} 0 \\ 0 \\ C' \end{bmatrix} = 0 .$$

This yields $w'C' = 0$, and since $C' \neq 0$ (not all the homogeneous coordinates may be zero, otherwise the point or line so defined is indeterminate) we have $w' = 0$; this is to say that all x' and y' ordinary coordinates that satisfy the equation are infinite.

Now $[w'x' \quad w'y' \quad w'] T = [wx \quad wy \quad w]$, and

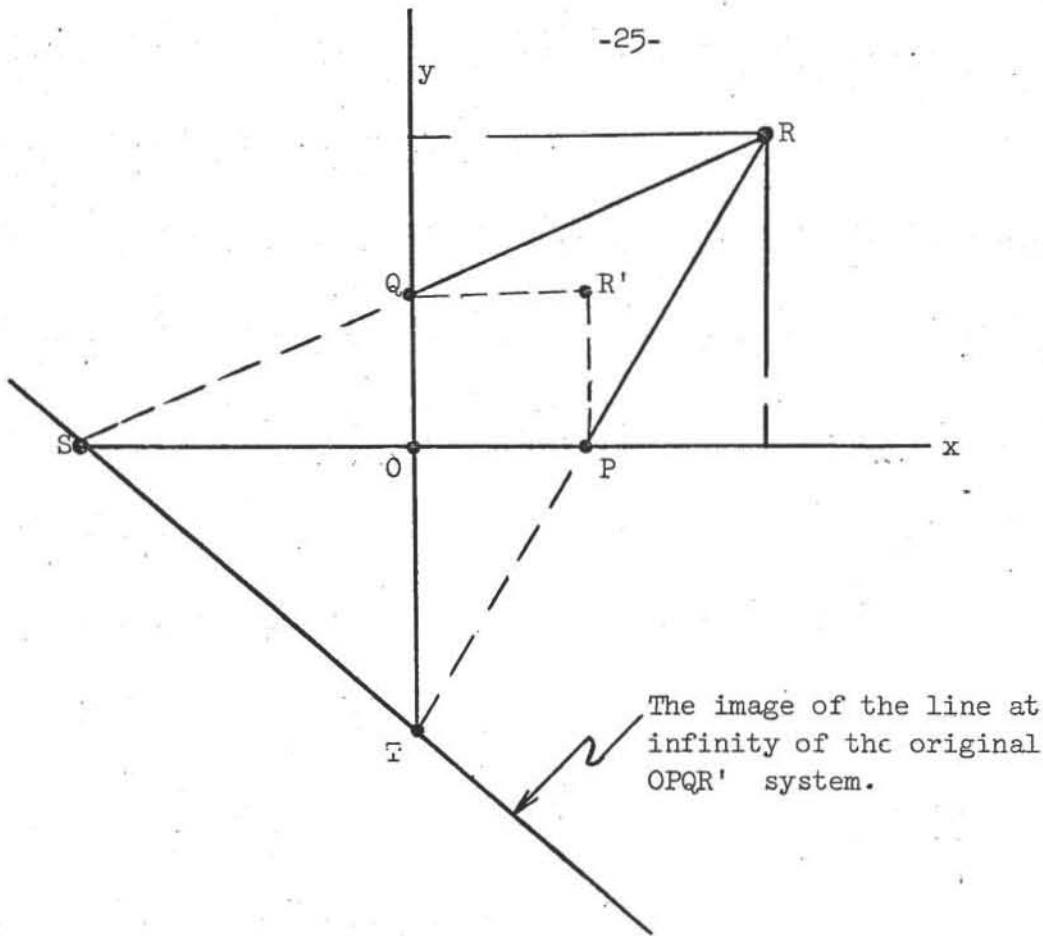
$$T^{-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \frac{1}{6} \begin{bmatrix} 3 & 0 & 1 \\ 0 & 3 & 1 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \frac{1}{6} \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} .$$

The equation of the transformation of the line at infinity in the original system is

$$[wx \quad wy \quad w] \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} = 0$$

or $[x \quad y \quad 1] \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} = 0$. This is evidently a local line, with equation

$x + y + 2 = 0$ or $y = -x - 2$. The figure showing the transformation will bring out several interesting points. The original four points are $OPQR'$ which have been transformed into $OPQR$. Note that before the transformation, lines OQ and PR' intersect at a point at infinity, since they are parallel. After the transformation, they are the lines OQ and PR , and intersect at the local point T . Similarly, OP and QR' intersect at infinity, but their images OP and QR intersect at the local point S . Indeed, S and T are the images of the infinitely distant points on the x and y axes, respectively. We have, for these points,



$$S \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 0 & -1 \\ 0 & 2 & -1 \\ 0 & 0 & 3 \end{bmatrix} = \begin{bmatrix} 2 & 0 & -1 \\ 0 & 2 & -1 \end{bmatrix}$$

Their ordinary coordinates are $\begin{bmatrix} -2 & 0 & 1 \\ 0 & -2 & 1 \end{bmatrix}$ and this is precisely what the linear equation requires, and what the figure shows.

FOUR DIMENSIONAL TRANSFORMATION AND THREE DIMENSIONAL PROJECTIVE TRANSFORMATIONS

We have seen, in the foregoing, that a 3 x 3 matrix can represent, on the one hand, a transformation of coordinates in three dimensions in which the origin remains fixed; but we have seen that if we regard one of the coordinate components of the point vector $[wx \ wy \ w]$ as a homogeneous coordinate, (say w). Then the 3 x 3 matrix represents a transformation of coordinates in two dimensions, and maps planes into planes.

The process of dividing the components of the vector by the chosen w coordinate is essentially equivalent to projection of the space, figure by rays or lines through the origin, followed by sectioning

the resulting bundle of rays by the plane $w = 1$. Thus the final result is a section of a three dimensional structure consisting of rays through the origin to all points of the three dimensional object.

We now extend this notion by an extra dimension. Consider the vector

$$[wx \quad wy \quad wz \quad w]$$

This vector can be thought of as descriptive of a point in four dimensional space, or it can be thought of as consisting of homogeneous coordinates descriptive of a point in three dimensional space, whose ordinary coordinates are obtained by projection and section. The ordinary coordinates are

$$\left[\begin{array}{cccc} \frac{wx}{w} & \frac{wy}{w} & \frac{wz}{w} & \frac{w}{w} \end{array} \right] = [x \quad y \quad z \quad 1]$$

Now $w = 1$ is no longer a plane, but is a section of a four dimensional space in which one of the degrees of freedom has been removed; it is therefore a three dimensional section of a four dimensional space.

Transformations are accomplished, as before, by multiplying point vectors by 4×4 matrices.

$$[x' \quad y' \quad z' \quad 1] \left[\begin{array}{ccc|c} a & b & c & 0 \\ d & e & f & 0 \\ g & h & i & 0 \\ k & l & m & 1 \end{array} \right] = [x \quad y \quad z \quad 1].$$

In this transformation, the upper left partition of the matrix contains nine numbers that describe shear and scale change transformations; the bottom row represents a translation of coordinates; the fourth column of the matrix has temporarily been specially chosen.

The matrix can be constructed by performing first the shear and scale change transformation, and then following this by the translation: this is shown by the matrix product

$$\begin{bmatrix} a & b & c & 0 \\ d & e & f & 0 \\ g & h & i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ k & l & m & 1 \end{bmatrix} = \begin{bmatrix} a & b & c & 0 \\ d & e & f & 0 \\ g & h & i & 0 \\ k & l & m & 1 \end{bmatrix}$$

We can now proceed to investigate the fourth column of the matrix, to see what effect entries in this column will have.

We take the matrix product

$$[x' \ y' \ z' \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & n \\ 0 & 0 & 0 & 1 \end{bmatrix} = [wx \ wy \ wz \ w]$$

When we perform the multiplication on the left, we obtain

$$[x' \ y' \ z'(nz' + 1)] = [wx \ wy \ wz \ w],$$

and division by $w = (nz' + 1)$ yields

$$[x \ y \ z \ 1] = \left[\frac{x'}{nz'+1} \ \frac{y'}{nz'+1} \ \frac{z'}{nz'+1} \ 1 \right].$$

This relates the new three dimensional coordinates $x \ y \ z$ to the original coordinates $x' \ y' \ z'$.

The transformation represents a mapping of one three-dimensional space into another. The mapping is accomplished by a transformation in four dimensions, followed by a projection and section to yield a three-dimensional space corresponding to $w = 1$.

Observe that the result is a three-dimensional space, and not a two-dimensional space.

We can make some qualitative remarks about the details of the transformation. The matrix

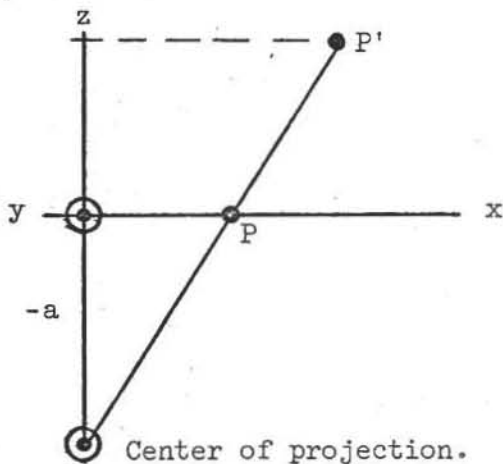
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

can as before be thought of as composed of four point vectors, describing, in homogeneous coordinates, the point at infinity on the x axis, the point at infinity on the y axis, the point at infinity on the z axis, and the origin of coordinates. Then, since

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & n \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & n \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

we see that three of these points are mapped into themselves; but the point at infinity on the z axis maps into the point $[0 \ 0 \ \frac{1}{n} \ 1]$; i.e., it becomes a local point.

Consider, from a different viewpoint, the projection of a point P' in an $x'y'z'$ coordinate system, into a point P in the plane $z' = 0$. We take the center of projection to be at $x' = 0$, $y' = 0$, $z' = -a$.



In the figure, the y axis appears as a point.

By similar triangles, we can obtain an expression for the ratio of the vectors from the center of projection to P and P' as follows

$$\frac{P}{a} = \frac{P'}{z'+a}$$

which leads directly to $P = \frac{a}{z'+a} P'$

$$= \frac{1}{\frac{z'}{a} + 1} P'$$

Now $P' = [x' \ y' \ z'+a]$ and $P = \left[\frac{x'}{\frac{z'}{a} + 1} \ \frac{y'}{\frac{z'}{a} + 1} \ a \right]$.

(Note that these vectors are from the center of projection to P' and P , not from the origin of coordinates.)

We see that if we set $\frac{1}{a} = n$, the matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & n \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

represents the transformation for the coordinates x and y . This is one possible interpretation of the matrix. We now see that it is descriptive of a perspective picture of the object point (or points) when imaged in the $z = 0$ plane; we might properly refer to this plane as the picture plane of the perspective construction.

Such a projection is sometimes called a "one-point" perspective, since all lines parallel to the z axis will appear to converge, in the picture, on the point $[x \ y] = [0 \ 0]$, the origin. This point is the so-called vanishing point of the picture. But lines not parallel to any of the axes will also have local vanishing points in the picture; for this reason, the term "one-point" perspective is somewhat misleading.

We can make the picture-projection process more general if we begin by performing a rotation on the object, followed by a translation, and finally perform the projective transformation just described.

A perspective pictorial that is in common use is the so called "two point" perspective, in which parallel horizontal lines converge in points on the "horizon line" of the picture. We shall examine such a perspective.

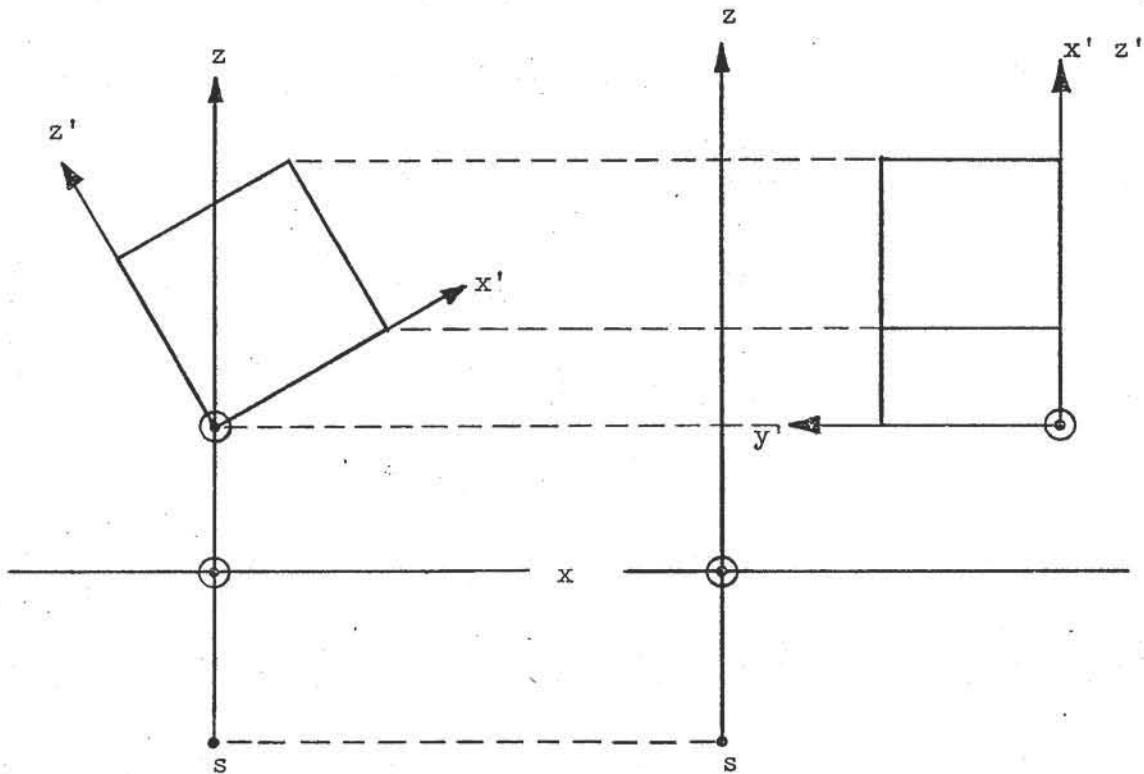
Suppose we wish to construct a picture of a unit cube, situated in space beyond the picture plane. We imagine the observer to be at point S , a unit in front of the $z = 0$ picture plane. This puts him at $z = -a$ on the z axis. The pure perspective transformation matrix is completely fixed by this number, as we have seen.

In order to perform the transformation, we need to establish the coordinates of the corners of the cube, and in order to do this, we need to define the cube in some way. We begin by attaching a coordinate system to the cube itself, preferably in such a way as to make it easy to describe the positions of the corners; then we establish the transformation that relates the cube coordinate system (call it the $[x' \ y' \ z' \ 1]$ system) to the observer's coordinate system $[x \ y \ z \ 1]$.

We might, for instance, attach a coordinate system to the cube so that three adjacent edges meeting in corner lie along the $x' \ y' \ z'$ axes. Then the coordinates of the corners of the cube may be written down immediately, as we already know.

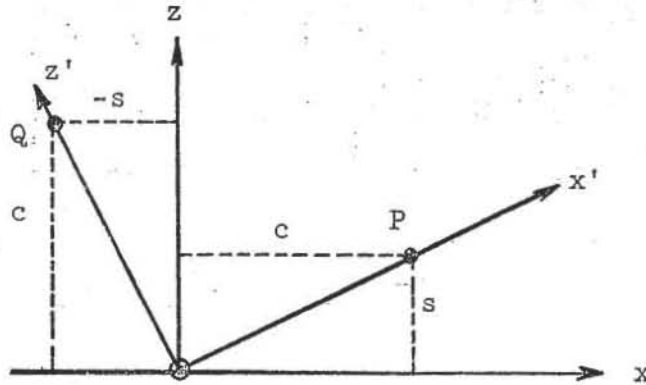
Suppose the observer is situated above the cube, looking down on it; and suppose the cube has been turned until its front face makes an angle of 30° to the picture plane, but the cube rests on a horizontal surface. This is the customary orientation for what is known as a "30°-60° two-point perspective."

A sketch will make clear the relative positions of observer and cube:



The cube coordinate system has been rotated 30° about the vertical y' axis, with respect to the observer's coordinate system. Then after the rotation it has been translated so that the original position of its origin, $[0 \ 0 \ 0]$, has become $[0 \ -2 \ 1]$ in $x \ y \ z$ coordinates.

The rotation matrix can be deduced from the new and old positions of unit points on the $x \ z$ and $x' \ z'$ axes, as follows (we neglect y and y' axes, since rotation takes place about them.):



$$\begin{bmatrix} P \\ Q \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ in the } x' y' \text{ system.}$$

In the $x y$ system, the matrix of coordinates becomes

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \text{ where } c \text{ and } s \text{ are the lengths of the projections of } OP \text{ and } OQ \text{ on the axes as shown.}$$

For the 30° rotation, $c = \cos 30^\circ$ and $s = \sin 30^\circ$. The rotation part of our matrix is now, for three dimensions:

$$\begin{bmatrix} c & 0 & s \\ 0 & 1 & 0 \\ -s & 0 & c \end{bmatrix}. \text{ Note that rotation about the } y \text{ axis causes}$$

the middle row and middle column of the matrix to have special values. Multiplication of $[x' y' z']$ by this matrix obviously yields a new x and a new z , but $y = y'$ and is unchanged.

If the origin of cube coordinates were translated to the point $[1 \ m \ n]$ in the observer's coordinate system, the translation part of the matrix would be this vector, and would become part of the bottom row of the transformation matrix.

In our case,

$$[1 \ m \ n] = [0 \ -2 \ 1].$$

For the perspective transformation, let the distance from the observer to the picture plane be $a = 1$. Then the combined matrix for the transformation is

We can also find the "vanishing points." These correspond to the points at infinity on the x' and z' axes; their matrix is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} c & 0 & s \\ 0 & 1 & 0 \\ -s & 0 & c \\ 0 & -2 & 2 \end{bmatrix} = \begin{bmatrix} c & 0 & s \\ -s & 0 & c \end{bmatrix}$$

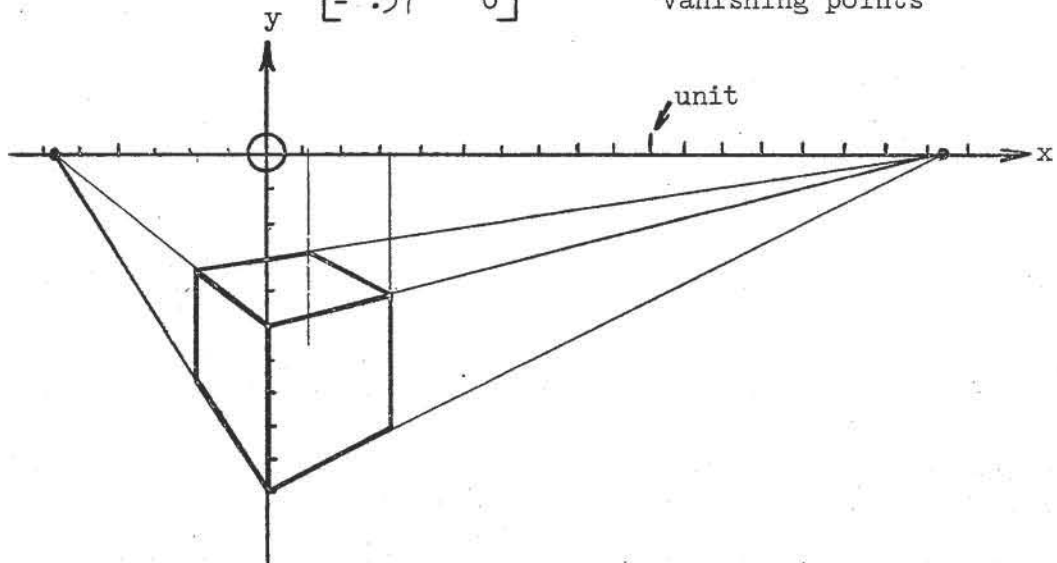
from which the ordinary coordinates turn out to be

$$\begin{bmatrix} \frac{c}{s} & 0 \\ -\frac{s}{c} & 0 \end{bmatrix} \text{ in the picture.}$$

Now $s = .5$, $c = .86$ for a 30° rotation. To two significant figures, the eight points of the cube and the two vanishing points have the following coordinates in the picture:

$$\begin{bmatrix} 0 & -1 & 0 \\ -.18 & -.70 & 1 \\ 0 & -.50 & 2 \\ -.18 & -.35 & 3 \\ .34 & -.80 & 4 \\ .11 & -.60 & 5 \\ .34 & -.40 & 6 \\ .11 & -.30 & 7 \end{bmatrix} \text{ cube corners}$$

$$\begin{bmatrix} -1.73 & 0 \\ -.57 & 0 \end{bmatrix} \text{ vanishing points}$$



The picture will, from an average viewing distance of ten inches, seem to be too tall for a cube. This is because the true viewing position for this picture is one unit directly in front of the origin of coordinates on this page. When viewed from this position, the vertical edges of the cube will appear properly foreshortened. Of course it is difficult to focus (or accommodate) the eye to such a short viewing distance, without the assistance of a magnifying glass. However, if the reader can find a glass with a focal length of about two or three inches, he can verify that the picture, viewed from the proper point, does indeed look like a cube.

THE THIN LENS EQUATION AND ITS ASSOCIATED PROJECTIVE TRANSFORMATION

For thin lenses, the equation

$$\frac{1}{z} + \frac{1}{z'} = \frac{1}{f}$$

describes the relationship of an object point at z' , to its image at z , in terms of the focal length of the lens, f . Measurements are made from the lens, for both z and z' . If we solve this equation for z as a function of z' , we obtain

$$\begin{aligned}\frac{1}{z} &= \frac{1}{f} - \frac{1}{z'} = \frac{z' - f}{fz'} \\ z &= \frac{fz'}{z' - f} = \frac{z'}{\frac{z'}{f} - 1}\end{aligned}$$

This expression is very similar to the expression found for our previous purely geometric interpretation of picture-making. In the optical case, we have also the relationships

$$\frac{x}{x'} = \frac{y}{y'} = \frac{z}{z'}$$

The matrix form of the photographic transformation is therefore

$$[x' \ y' \ z' \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & +\frac{1}{f} \\ 0 & 0 & 0 & -1 \end{bmatrix} = [wx \ wy \ wz \ w]$$

From this,

$$[wx \quad wy \quad wz \quad w] = [x' \quad y' \quad z' \quad (\frac{z'}{f} - 1)]$$

or

$$[w \quad y \quad z \quad 1] = \begin{bmatrix} \frac{x'}{\frac{z'}{f} - 1} & \frac{y'}{\frac{z'}{f} - 1} & \frac{z'}{\frac{z'}{f} - 1} & 1 \end{bmatrix} .$$

It is interesting to see the limits of the object space compared to the limits of the image space

Image Space	Object Space
z	z'
0	0
∞	f
f	∞

This table presents the well-known conjugate focal plane behavior of lenses. Now return to the transformation represented by our previous derived matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{a} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

From this, $z = \frac{z'}{\frac{z'}{a} + 1}$.

Here, object-image space comparison table may similarly be constructed:

Image Space	Object Space
z	z'
0	0
$\frac{a}{2}$	a
a	∞

In this case, all of the positive half-space in the object domain is imaged in the finite band between 0 and a .

Both of these transformations are sometimes called "relief perspectives." We usually think of the photographic process as producing a plane image of a three-dimensional space, but a moment's reflection reminds us that we must focus cameras; this implies that we must put the

photographic film plane in the proper position in the image space to correspond to a particular plane in object space. This is a physical confirmation of the remark made earlier, that the general projective transformation images 3-space into another 3-space.

There is however a special transformation that carries 3-space into the 2-space of a plane. It is represented by the matrix product

$$[x' \ y' \ z' \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} = [wx \ wy \ wz \ w] .$$

Then, performing the multiplication,

$$[x' \ y' \ z' \ z'] = [wx \ wy \ wz \ w] .$$

We see that $w = z'$, and

$$x = \frac{x'}{z'} \quad y = \frac{y'}{z'} \quad z = \frac{z'}{z'} = 1 .$$

The matrix has a row of zeros. Therefore its determinant vanishes, and it has no inverse. This is to say that once the transformation has occurred, there is no way to obtain 3-dimensional information back again from the plane figure. But this is obvious.

AXONOMETRY

A special case of the projective transformation matrix puts the projection point at infinity. Then the picture transformation becomes simply the identity matrix, since $\frac{1}{a} = 0$. We need pay attention only to the rotation-translation part of the transformation. Furthermore, the translation part of this matrix merely serves to move the origin, and in this case it becomes a trivial part of the transformation. We neglect it, and pay attention only to the rotation part of the matrix. We can omit the fourth homogeneous coordinate, and consider only such expressions as

$$[x' \ y' \ z'] \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = [x \ y \ z] .$$

Plane projections can be obtained from this relationship by plotting $[x y]$ $[x z]$ or $[y z]$.

The transformation and the associated plane projections include all possible cases of what is known as "parallel projection," and in descriptive geometry this is called "axonometry;" the pictures of an object made in this way are called "axonometric projections."

The general class of axonometric projections break down into a number of special categories:

TRI-METRIC projections, in which the transformation matrix is a pure rotation, and orthogonality of the transformed axes is preserved.

DI-METRIC projection is a special case of tri-metric projections, in which two of the axes are equally foreshortened.

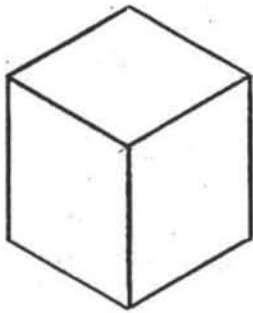
ISOMETRIC projection is a special case of di-metric projection, in which all three axes are equally foreshortened. This of course leads to a unique matrix. The other two more general cases have certain arbitrary characteristics.

OBLIQUE PROJECTIONS, in which the transformation matrix no longer preserved orthogonality of the coordinate axes.

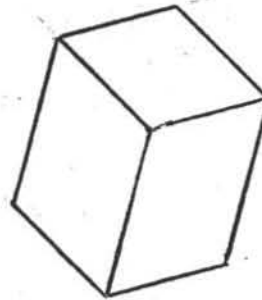
CAVALIER projection is a special case, in which two axes appear perpendicular in the picture, and are not foreshortened; the third axis is inclined with respect to the horizontal axis and is not foreshortened.

CABINET projection is a special case of the cavalier projection; foreshortening by a factor of $1/2$ occurs to lines parallel to the third axis.

In order to preserve orthogonality of the axes in space, the matrix must represent a pure rotation. Ordinarily, in engineering use, drawings of objects preserve verticals of objects as verticals on the drawing. We see pictures of rectangular objects as

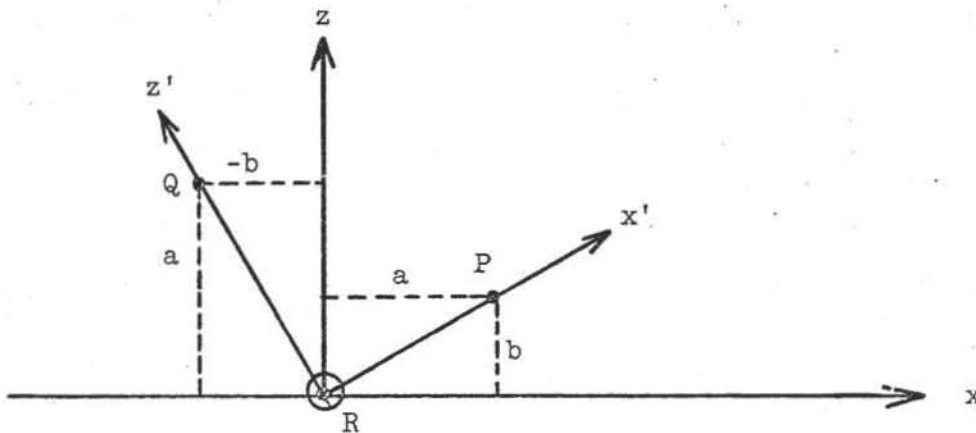


rather than



unless we really mean to have the object tipped.

We can obtain the full rotation matrix, subject to this restriction, by two simple rotations compounded, as follows.



We rotate first about the vertical y axis; the matrix is, by inspection,

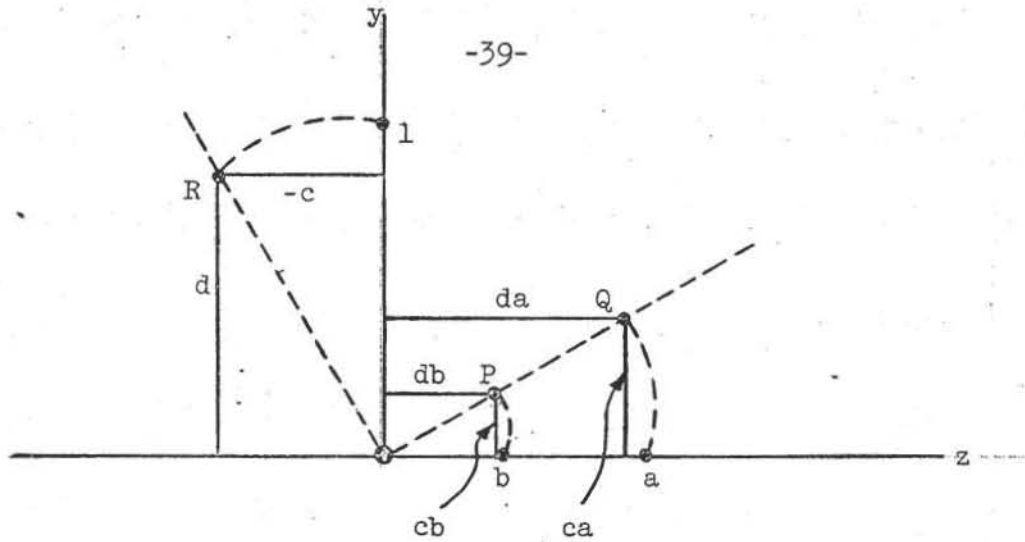
$$\begin{bmatrix} a & 0 & b \\ 0 & 1 & 0 \\ -b & 0 & a \end{bmatrix}$$

as we have seen in our previous discussion of projective

transformations. The unit points P and Q now have new coordinates in the observer's coordinate system $[x \ y \ z]$.

We next propose to rotate the resulting figure about the x axis (Not the x' axis.)

We take a side view of the state of affairs after the first rotation:



Before the second rotation, the three unit points have coordinates

$$\begin{bmatrix} a & 0 & b \\ 0 & 1 & 0 \\ -b & 0 & a \end{bmatrix} = \begin{bmatrix} P \\ R \\ Q \end{bmatrix}$$

After the rotation, these points have coordinates which remain unchanged in x (since rotation takes place about this axis) but change in y and z .

The new coordinates of R are, by inspection,

$$[0 \quad d \quad -c]$$

Similarly, the new coordinates of P are

$$[a \quad bc \quad bd]$$

and the new coordinates of Q are

$$[-b \quad ac \quad ad]$$

This is the result of a second simple rotation, represented by the matrix

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & d & -c \\ 0 & c & d \end{bmatrix}$$

Indeed, $\begin{bmatrix} a & 0 & b \\ 0 & 1 & 0 \\ -b & 0 & a \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & d & -c \\ 0 & c & d \end{bmatrix} = \begin{bmatrix} a & bc & bd \\ 0 & d & -c \\ -b & ac & ad \end{bmatrix}$ as we might have written

immediately. Note the zero in the first column of the matrix. This represents the transformed x coordinate of the unit point R on the vertical y axis; it remains on the vertical y axis during the transformation, and the condition on vertical lines is thus satisfied.

The individual rotations preserve orthogonality, and if both $a^2 + b^2 = 1$, and also $c^2 + d^2 = 1$, then they preserve the size of the object as well. Consequently the combined matrix should have the property that lengths are preserved after the transformation.

Consider the length of the vector from the origin to the unit point, P, and the length of this same vector after rotation.

We have, for this length, the vector "scalar" product

$$\begin{bmatrix} a & bc & bd \end{bmatrix} \begin{bmatrix} a \\ bc \\ bd \end{bmatrix} = a^2 + b^2c^2 + b^2d^2 = 1^2 .$$
$$= a^2 + b^2(c^2+d^2) = a^2 + b^2 = 1$$

which shows that this length is unchanged.

DI-METRIC PROJECTIONS

As we have noted, the rotation transformation yields the class of axonometric projections known as TRI-METRIC projections; all auxiliary views obtained in engineering descriptive geometry are trimetric projections. It is well known that any view of an object can be obtained in descriptive geometry by two auxiliary views. This amounts to performing two rotations on the object.

If we impose the condition for a di-metric projection, (to make two of the axes equally foreshortened) we must have equal lengths for the x y projections of the unit point vectors on the chosen axes. The matrix of coordinates of the unit points is

$$\begin{bmatrix} a & bc & bd \\ 0 & d & -c \\ -b & ac & ad \end{bmatrix} \begin{matrix} x \text{ axis unit point} \\ y \text{ axis unit point} \\ z \text{ axis unit point} \end{matrix}$$

If we select x and y axes to be equally foreshortened, we can write the equation of lengths of these axes in the projection:

$$a^2 + b^2c^2 = d^2 .$$

But $a^2 = 1 - b^2$ and $d^2 = 1 - c^2$. Substituting and carrying out a little algebra, we get an expression for b in terms of c :

$$b^2 = \frac{c^2}{1-c^2} .$$

We can always choose c arbitrarily, and then b , a , and d can be found, thus completely defining the matrix for the di-metric projection.

In one such very commonly used projection, an additional requirement is that the third, z axis, shall be foreshortened by a factor of $1/2$. This implies that, for this axis,

$$b^2 + a^2c^2 = \frac{1}{4} .$$

This is sufficient, when combined with the previous equation, to yield, after a little algebra,

$$c^2 = \frac{1}{8} .$$

Using this, the entire matrix is completely and uniquely defined.

ISOMETRIC PROJECTIONS

Another very much used special case of axonometry is the one in which all three axes are equally foreshortened. (TRI-METRIC projections are often innocently referred to as "ISOMETRIC" projections, even by some engineers who ought to know better. It makes understanding of exactly what is being discussed a little difficult.)

If we impose this condition on the projected lengths of the unit vectors on the axes, we have, for the dimetric,

$$b^2(1-c^2) = c^2$$

as already determined, and the new condition for the z axis:

$$b^2 + a^2c^2 = d^2$$

After a little algebra, we can combine these two equations to learn that

$$c^2 = \frac{1}{3} , \text{ or } c = \frac{\sqrt{3}}{3} .$$

Again the transformation matrix is completely and uniquely defined. The projected x axis makes an angle with the "horizontal" axis of the picture coordinate system defined by

$$\text{TAN } \alpha = \frac{bc}{a} .$$

It is easy to find that $b^2 = a^2 = \frac{1}{2}$, so that $\text{TAN } \alpha = \sqrt{c} = \frac{\sqrt{3}}{3}$:

Hence $\alpha = 30^\circ$, a very well known result. The projection is ISO-METRIC. All three axes have equal scales.

OBLIQUE PROJECTIONS (Cabinet and Cavalier)

The requirements here for these two special cases of oblique projections is that one pair of the axes (say x and y) remain mutually perpendicular, and not foreshortened. The third z axis is to make an angle of 45° with the "horizontal" and is to be foreshortened by a factor m . ($=1$ or $\frac{1}{2}$).

The matrix of the transformation is, of course, simply

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & m & 0 \end{bmatrix} .$$

The third column of the matrix is immaterial, since we are not going to use it.

This last matrix is of course trivial--it is scarcely necessary to use it to compute points in the transformation; nevertheless it has been exhibited to show that all these special cases fall under the general theory.

If the picture is being plotted in $x y$ coordinates, and the transformed z is simply ignored, this is equivalent to multiplying the transformation matrix by a projection matrix:

$$[x \ y \ z]A = [x' \ y' \ z']$$

and then

$$[x' \ y' \ z'] P = [x' \ y' \ 0]$$

$$AP = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} a & b & 0 \\ d & e & 0 \\ g & h & 0 \end{bmatrix}$$

or

$$[x \ y \ z] \begin{bmatrix} a & b \\ d & e \\ g & h \end{bmatrix} = [x' \ y'] .$$

Strictly speaking, this is the correct interpretation of the axonometric projections. Note that the matrix which is the product of AP has a vanishing determinant; hence, as we might suppose from geometric considerations, it has no inverse; we cannot regain space information about an object from a single view.

Algebraic geometry has a theorem about axonometric projections:

Every affine transformation with vanishing determinant is an axonometric projection.

The word "affine" refers to those transformations which can be described by any 3 x 3 matrix.

TRANSFORMATIONS BY COMPUTER

The foregoing discussion has been carried out in detail, and much of the detail is superfluous when we perform transformations by computer. For instance, the oblique projections (cavalier, cabinet, and others) were devised for convenience in the drafting room; they make it easy to construct, from working drawings of objects, pictures that are more easily understood by the uninitiated than the working drawings themselves. But they are not very good pictures. Isometrics are better, and again are very easily constructed by graphical procedures. Dimetrics are better still, and of the entire class, trimetrics, the hardest to construct, are the best representations short of perspective pictures. But the computer can construct a tri-metric picture of an object as easily as any of the less desirable forms, and it is possible to implement the matrix loading and multiplication so that the object can be rotated in the picture in "real-time."

Perspective pictorials take a little longer, since the homogeneous coordinate must be divided out, and division is ordinarily a long process, relative to addition and multiplication.

This concludes the discussion of transformations.

A P P E N D I X T W O

HOMOGENEOUS MATRIX REPRESENTATION AND MANIPULATION
OF N-DIMENSIONAL CONSTRUCTS

Lawrence G. Roberts
Lincoln Laboratory, Massachusetts Institute of Technology
Lexington, Massachusetts

May 1965

MS-1405

Operated with support from the U.S. Air Force.

HOMOGENEOUS MATRIX REPRESENTATION AND MANIPULATION OF N-DIMENSIONAL CONSTRUCTS

Lawrence G. Roberts

Lincoln Laboratory,* Massachusetts Institute of Technology
Lexington, Massachusetts

The representation and processing of graphical information has been found to be greatly simplified if a system of homogeneous coordinates is used in conjunction with the appropriate matrix techniques. The following notes are an attempt to set down the matrix forms and methods which have been found useful in the representation and display of graphical data. This work is an extension of material already presented as a thesis.¹

The specific items of interest are points, lines, conic sections, planes, and quadric surfaces in two and three dimensions. However, the techniques are not limited by either the dimension or order of the space, and can be extended in a straightforward manner. The use of homogeneous coordinates throughout is extremely important in order to maintain the simplicity of the results although its original purpose was to allow perspective transformations. It is assumed that in any graphical system there exists a data structure (such as the CORAL list structure used at Lincoln Laboratory) in addition to the matrices which contain the information about the associations between elements. This structure is a separate subject and will not be discussed herein.

Homogeneous Coordinates

The homogeneous coordinate technique is simply the representation of n-space objects in $(n+1)$ - space in such a way that a particular perspective projection recreates the n-space. It can also be thought of as the addition

* Operated with support from the U.S. Air Force

of an extra coordinate to each vector, a scale factor, so that the vector has the same meaning after multiplication by a constant. For example, in 2-D a point $[a, b]$ would be entered as $[a, b, 1]$ and then manipulated as a 3-D vector. For display of a point, the 3-D vector $[x, y, w]$ would be transformed back to 2-D by:

$$a = x/w \quad , \quad b = y/w$$

Thus, the w component is a scale factor and can often be thought of as a dependent variable.

The 3-space created by a homogeneous treatment of 2-D points and lines consists of lines and planes all passing through the origin. Thus, a single 3×3 transformation matrix may be used to rotate and translate the points in 2-space as well as allowing perspective transformations.

Notation

A consistent notation will be utilized throughout to minimize the number of comments required. The following conventions will be followed:

Matrices:	Always capital letters but sometimes with subscripts.
Vectors:	row vectors: p, r, v column vectors: γ, λ
Single Variables:	free parametric variables: s, t specific coordinates: x, y, z, w other variables: a, b, c, d, e, f, g, h

The transpose of A is written A' .

2-D Point and Line Representation

A. A point in the space is a 3-element row vector:

$$v = [x, y, w]$$

THE POINT $X = \frac{x}{w}$ $Y = \frac{y}{w}$
IS PROJECTION OF (x, y, w) ON
 $w=1$ PLANE

Any constant multiple of v represents the same point.

B. A line in the space is represented by a column vector:

$$Y = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

VECTOR IS NORMAL TO
PLANE THROUGH LINE AND
ORIGIN.

C. The scalar product of a point and a line produces a number which is zero if the point is on the line, minus if it is on the side and plus if it is on the other side.

DOT PRODUCT = ZERO ON LINE!

Line Equation: $v_Y = 0$ ($ax + by + cw = 0$)

D. The distance from a point to a line is indicated by the product v_Y but must be normalized if absolute distance is required:

Distance from v to Y : $d = (v_Y) / w \sqrt{a^2 + b^2}$

$\frac{1}{w} = \text{NORMALIZE FOR } v$

$\frac{1}{\sqrt{a^2 + b^2}} = \text{NORMALIZE FOR } Y$

This distance is still a signed quantity indicating which side of the line the point is on.

E. A transformation H of the space is a 3×3 matrix.

The transformed point (v_1) is obtained: $v_1 = vH$

The transformed line (Y_1) is obtained: $Y_1 = H^{-1} Y$

Thus the line equation is unchanged: $v_1 Y_1 = vY = 0$

$H = H^{-1}$

2-D Points and Lines

A. Find line γ given two points v_0 and v_1 :

$$v_0 = [x_0, y_0, w_0] \quad , \quad v_1 = [x_1, y_1, w_1]$$

$$\gamma' = [(y_1 w_0 - y_0 w_1), (x_0 w_1 - x_1 w_0), (x_1 y_0 - x_0 y_1)]$$

$\gamma' = v_0 \times v_1$ (cross product)

B. Find intersection v given two lines γ_0 and γ_1 :

$$v = \gamma_0 \times \gamma_1 \text{ (cross product)}$$

$$\gamma_0' = [a_0, b_0, c_0] \quad \gamma_1' = [a_1, b_1, c_1]$$

$$v = [(b_1 c_0 - b_0 c_1), (a_0 c_1 - a_1 c_0), (a_1 b_0 - a_0 b_1)]$$

C. Find transformation T which translates space so point v becomes the origin:

$$v = [x, y, w]$$

$$T(v) = \begin{bmatrix} w & & \\ & w & \\ -x & -y & w \end{bmatrix} \quad v T(v) = (wx - wx), (wy - wy)$$

I would be BETTER

D. Find line γ which is normal to line λ and passes through point v :

1. prepare $T(v)$

2. $\gamma = T(v) K \lambda$

$$K = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$K \lambda = \begin{bmatrix} a \\ -b \\ 0 \end{bmatrix} \quad \gamma = -wa \quad (-bx + ay)$$

E. Find line γ which is parallel to line λ and passes through point v :

$$\gamma = T(v) K^2 \lambda$$

$$K^2 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

4
 K^2 ... w component

OR

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{will do}$$

$K^2 = K$

can make g, h, d
and divide by g
into b, d, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z

Note: If $v \cdot Cv = 0$
Then multiples of v do.
So it's conical at most.
and where $w=1$ is

$$\begin{bmatrix} a & e & h \\ c & f & i \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

$$ax^2 + (b+d)xy + ey^2 + (g+c)xw + (h+f)yw + iw^2 = 0$$

- 2-D Conic Representations

- A. Implicit Conic Representation $vCv' = 0$ $C: (3 \times 3)$
- B. Parametric Conic Segment Representation $v = rA$ $A: (3 \times 3)$
free parameter t $r = [t^2, t, 1]$

1. Since the terms of r are dependent on t there is a unique quadratic relation which states the constraint on r .

$rKr' = 0$ where: $K = \begin{bmatrix} 1 & & \\ & -2 & \\ & & 1 \end{bmatrix}$

$rK = [1, -2t, t^2]$
 $w, -2y, x$

or $w = y^2$

2. Due to relation (1) it is possible to determine C from the parametric form.

$$C = A^{-1} K A^{-1}$$

$$C^{-1} = A' K^{-1} A$$

3. Other parametric forms. It is often useful to use some transformation J of the parametric vector r in the following way:

$v = pB$ where: $p = rJ$, $B = J^{-1} A$

then: $pMp' = 0$ where $M = J^{-1} K J^{-1}$

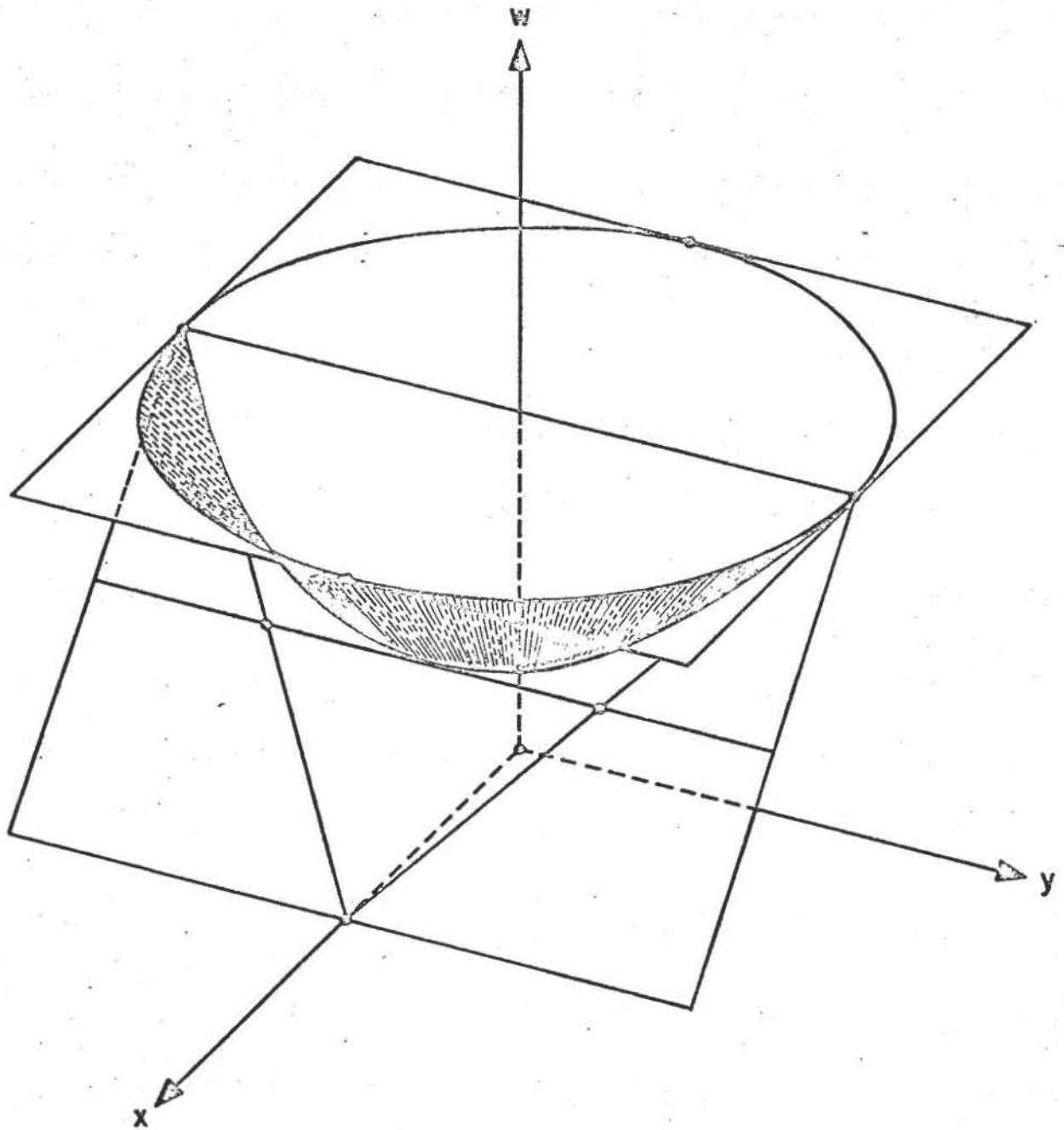
thus: $C = B^{-1} M B^{-1}$

4. Tangent Point Transform: A useful parametric form is one in which the rows of B are the start point, end point, and the intersection point of the tangents from these points.

$v = pB$ $J_1 = \begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -2 \\ 0 & 0 & 1 \end{bmatrix}$ $M_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -\frac{1}{2} & 0 \\ 1 & 0 & 0 \end{bmatrix}$

$P_1 = [t^2, t, 1]$

Also (Cv') is \perp to v since $vCv' = 0$.



Projection of Parabola Into a Circle in a Homogeneous Coordinate System.

The parabola on the slanted surface is used as a parametric representation of the circle. The projection is from the origin onto the $w = 1$ plane.

$$k^2(v_t C v_t') = -\frac{1}{2}(v_1 C v_1')$$

thus: $k = \pm \sqrt{-\frac{(v_1 Y_1)}{2(v_t C v_t')}}}$

(sign determines segment)

Now:

$$B = \begin{bmatrix} v_1 \\ kv_t \\ v_0 \end{bmatrix}$$

d. If we want the basic parametric form:

$$A = J_1 B$$

- e. NOTES: 1. The relative scale of v_0 and v_1 is free but affects the speed of movement along the arc. For minimum velocity change along the arc they should be scaled so that $w_0 = w_1$ before starting the process. ($v = [x, y, w]$)
2. Parallel tangents work without problems. ($w_t = 0$)

2-D Conic, Point, Line Relationships

A. The tangent line Y to a conic C at a point v on the conic is given by:

$$Y = C v'$$

B. The polar line Y of a pole v with respect to a conic C is related just as above:

$$Y = C v'$$

However, when v is not on the conic the polar line found has the following properties:

1. γ has the same slope as the conic at the conic's nearest point to v . That is it is perpendicular to the normal from v to C .
2. γ also intersects C at such points where tangent lines can be drawn to C through v .

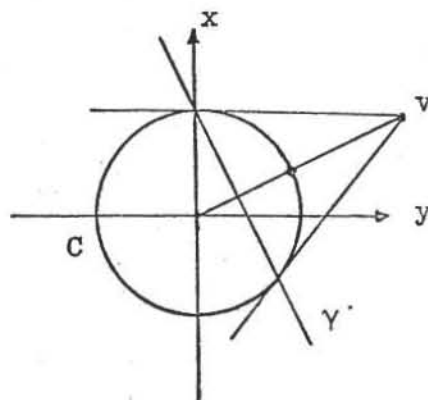
not true
Darny conic

Example:

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$v = [1 \ 2 \ 1]$$

$$\gamma = Cv' = [1 \ 2 \ -1]'$$



3. The polar lines of all points on γ go through v .

C. When the matrix C has an inverse C^{-1} (i.e., it represents a curve, not a line) then the pole may be found given a polar line γ .

Pole: $v = \gamma' C^{-1}$

1. The pole of the line at infinity is the center of the conic: $v = [0 \ 0 \ 1] C^{-1}$
2. Given a line tangent to the conic then the pole is the point of contact.

D. Intersection of a line γ and a conic C : Find the intersection points v_j (0, 1, or 2):

1. Prepare a parametric matrix for the line γ such that $v = [t, 1] P$ lies on the line for all t .

Now: $\gamma = [a \ b \ c]$

If $a \neq 0$ or If $b \neq 0$ or If $a=b=0$

$$P = \begin{bmatrix} -b & a & 0 \\ -c & 0 & a \end{bmatrix}$$

$$P = \begin{bmatrix} -b & a & 0 \\ 0 & -c & b \end{bmatrix}$$

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

2. Now compute the terms of the quadratic equation:

$$PCP' = \begin{matrix} a & b \\ b & c \end{matrix} \quad (at^2 + 2bt + c = 0)$$

a. if $a = 0$ then $v = [-c, 2b] P$

b. otherwise: $k_j = -b \pm \sqrt{b^2 - ac}$ ($b^2 < ac \Rightarrow$ no intersections)

and: $v_j = [k_j, a] P$ (two solutions)

E. Intersection of two conics:

Represent one conic as parametric: $v = r A$

Represent other conic as implicit: $vCv' = 0$

Compute:

$$ACA' = \begin{bmatrix} a & b & c \\ b & d & e \\ c & e & f \end{bmatrix}$$

Solve: $at^4 + 2bt^3 + (2c+d)t^3 + et + f = 0$ for t_j

Now: $v_j = r_j A$ where $r_j = [t_j^2, t_j, 1]$

3-D Representations

A. A point vector is as before but has one more coordinate:

$$v = [x, y, z, w]$$

B. A line or space curve can be represented;

1. Parametricly as before:

$$v = r A \quad \text{where } r = [t^2, t, 1] \text{ but } A \text{ is } 4 \times 3$$

If higher than second order space curves are wanted, r can be extended to include more terms and A extended likewise.

2. As the intersection of two planes or a plane and a quadric surface. However, it is difficult to use this representation directly.

C. A plane is now the item represented by a column vector:

$$Y = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

and a plane equation is: $vY = 0$

D. Any quadric surface (including planes) can be represented by a quadratic form as was used before for 2-D conics. Here the matrix F will be a 4 x 4.

$$vFv' = 0$$

There are four canonic forms of quadric surfaces under perspective transformation. These are the groups in which some transform H will transform F_1 into F_2 as below (congruence):

$$F_2 = H F_1 H'$$

Groups:

1. Sphere, ellipsoid, elliptic paraboloid, hyperboloid of two sheets.
2. Hyperboloid of one sheet, hyperbolic paraboloid.
3. Cone, cylinder, hyperbolic sheets, parabolic sheet (Rank 3)
4. Intersecting, parallel, and single planes (Rank 2)

Most calculations in 3-D are the same as they were in 2-D except that the dimension of the vectors and matrices has increased. Also, it must be remembered that where lines and curves were considered before the corresponding manipulations are now for planes and quadric surfaces.

3-D Coordinate Transformations

Given a 4 x 4 transformation H:

A. Points: $v_1 = vH$

B. Planes: $\gamma_1 = H^{-1} \gamma$

C. Quadrics: $F_1 = H^{-1} F H^{-1}$

3-D Planes and Points

A. The intersection of 3 planes $\gamma_1, \gamma_2,$ and γ_3 is a point v .

$$v = \left[\begin{array}{c} \left| \begin{array}{ccc} d_1 & b_1 & c_1 \\ d_2 & b_2 & c_2 \\ d_3 & b_3 & c_3 \end{array} \right|, \left| \begin{array}{ccc} a_1 & d_1 & c_1 \\ a_2 & d_2 & c_2 \\ a_3 & d_3 & c_3 \end{array} \right|, \left| \begin{array}{ccc} a_1 & b_1 & d_1 \\ a_2 & b_2 & d_2 \\ a_3 & b_3 & d_3 \end{array} \right|, \left| \begin{array}{ccc} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{array} \right| \end{array} \right]$$

B. The plane through 3 points is just as above except for the interchange of (x, y, z, w) for (a, b, c, d) and γ' for v .

C. The distance from a point v to a plane γ is:

$$d = (v\gamma) / w \sqrt{a^2 + b^2 + c^2}$$

D. A plane γ parallel to plane λ through point v :

$$\gamma = T(v) J_\lambda \quad T(v) = \begin{bmatrix} w & 0 & 0 & 0 \\ 0 & w & 0 & 0 \\ 0 & 0 & w & 0 \\ -x & -y & -z & w \end{bmatrix} \quad J = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

3-D Quadric Surfaces

- A. The plane γ tangent to a quadric F at a point v on the surface is:

$$\gamma = Fv'$$

- B. As before the above relation also defines the polar plane to a point.

1. A polar plane cuts the quadric surface in a conic section which is the outline of the quadric as seen from the pole v . This is very important because we want to be able to display this outline.

2. The polar plane is also perpendicular to the normal from v to the quadric surface.

- C. Intersection of a quadric F and a plane γ :

Since we normally wish to display the intersections we find it is convenient to find the projection of the intersection as seen from $x = \infty$ which we assume has been transformed to be the viewing point. Thus, we will find a 2-D conic matrix C as a result of the intersection.

1. Prepare: $P = \begin{bmatrix} -b & a & 0 & 0 \\ -c & 0 & a & 0 \\ -d & 0 & 0 & a \end{bmatrix}$ where $\gamma' = [a \ b \ c \ d]$

2. Now: $C = PFP'$

(Good unless $a = 0$ in which case projection is line $\lambda = [b \ c \ d]$)

D. Outline of Quadric from $x = \infty$:

Since the pole $x = \infty$ has a polar plane: $Y = F [1, 0, 0, 0]$, we can easily cut F with this plane which is its own first column to obtain the conic C describing its outline. This can proceed just as in part C above.

There happens to be a particular simplification when the intersecting plane is a column from F.

Assume first column of F is: $Y^t = [a \ b \ c \ d]$

$$\text{Then: } Q = MF \quad \text{where} \quad M = a \begin{bmatrix} 0 & 0 & 0 & 0 \\ -b & a & c & 0 \\ -c & 0 & a & 0 \\ -d & 0 & 0 & a \end{bmatrix}$$

This computation will leave the first column and top row of Q all zeros with the lower right 3×3 being C. Really Q is the quadric surface normal to the $x = 0$ plane which is just tangent to the quadric F.

If $a = c$: and $b = c = d = 0$ then $Q = F$, otherwise: no outline visible.

Volume Representation

In order to represent solid objects a group of planes or quadric surfaces can be used to bound the desired volume. Since the homogeneous coordinate system allows each plane vector or quadric matrix to be multiplied by an arbitrary constant, the sign of each surface can be adjusted so as to produce a positive product for points inside the volume and negative for outside.

Point v inside plane γ : $v\gamma > 0$

Point v inside quadric F: $vFv' > 0$

This technique has been used previously¹ where all the plane vectors of a convex solid were grouped into a "volume" matrix. Thus, when a point was multiplied by this matrix the resultant vector was all positive if the point was inside the volume. To extend this concept to quadric surfaces requires a volume tensor, that is a set of matrices. Further, to represent complex volumes there needs to be several "volume" tests and a Boolean combination of the results. The particular form such a combination should take may well depend on the problem to be solved.

Hidden line elimination for displaying groups of objects is one of the prime reasons for the representation of volumes and was previously worked out for plane surfaced objects. For curved surfaced objects the manipulation techniques presented in this paper are sufficient as long as quadric surfaces are only allowed to intersect planar surfaces. However, the math takes a quantum jump in complexity when fourth-order space curves are introduced by quadric-quadric intersections. Until there is a demonstrated need for such solutions in practice, the large investment of time and effort required is probably unwarranted.

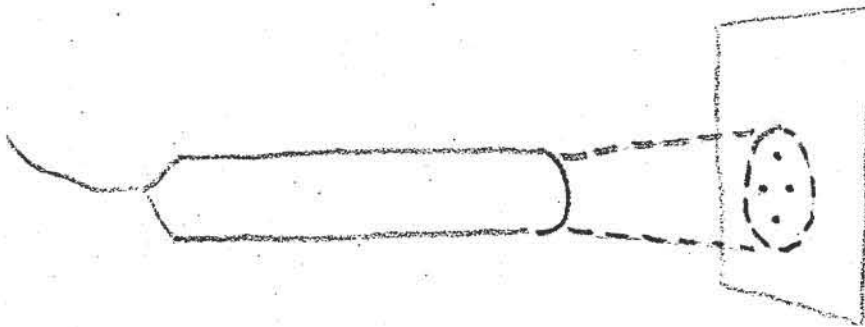
1. "Machine Perception of Three-Dimensional Solids", Lincoln Laboratory Technical Report, #315, (22 May 1963). L. G. Roberts.

A P P E N D I X I I I

P E N T R A C K I N G

In order to get position information from a light pen it is necessary to provide a program or hardware which follows changes in the position of the pen. Such a system, called "pen tracking" is a relatively old technique. In spite of its age and susceptibility to analysis, pen tracking remains a considerable mystery to many people. It is my hope in this appendix, drawn entirely from discussion with Tom Stockham, to dispell some of the mystery associated with pen tracking. In some sense, however, the points made here are moot because it appears that the light pen is rapidly being replaced by stylus input devices and comparitors.

The original tracking systems followed motion of the light pen by means of a pattern of four dots.



By sensing which of these dots was visible to the pen, the tracking program could discover which direction to move the pattern in order to remain within the field of view. Such a pattern, called a "tracking square", or "tracking diamond" was well known by 1957.

The algorithm for manipulating a tracking square is relatively simple. If none of the points fall in the field of view of the light pen, the array will be left alone. If one or more of the points are seen by the light pen, then the entire array will be moved in such a direction as to bring all of the points within the field of view; that is, if the right-hand point only is seen, then the array will be moved to the right. If all points except the bottom one are seen, then the array will be moved up, and so on. A flow chart to do that might be as shown in the figure on the following page. (Figure 2.)

Such a tracking program responds to a light pen hit on one of the four points of the tracking square by branching to a location in which the position of the square is moved. The desire to branch conditionally on a hit after each point of the tracking figure is posted is in direct conflict with the "normal" pointing function of the light pen in which the address in memory of the item which was seen is recorded for use at the end of the frame. Pen tracking programs commonly prefer the "hit flag" light pen logic rather than the interrupt type.

Suppose, on the other hand, that we wish to use the interrupt type of light-pen hardware. In such a case, we can treat the tracking cross as merely a part of the display list. The light pen interrupt routine will indicate which of the dots in the tracking square were or were not seen. At the end of the frame, a new location for the tracking square can be computed from that information. Such an arrangement would be as shown in the Figure on the following page. (Figure 3)

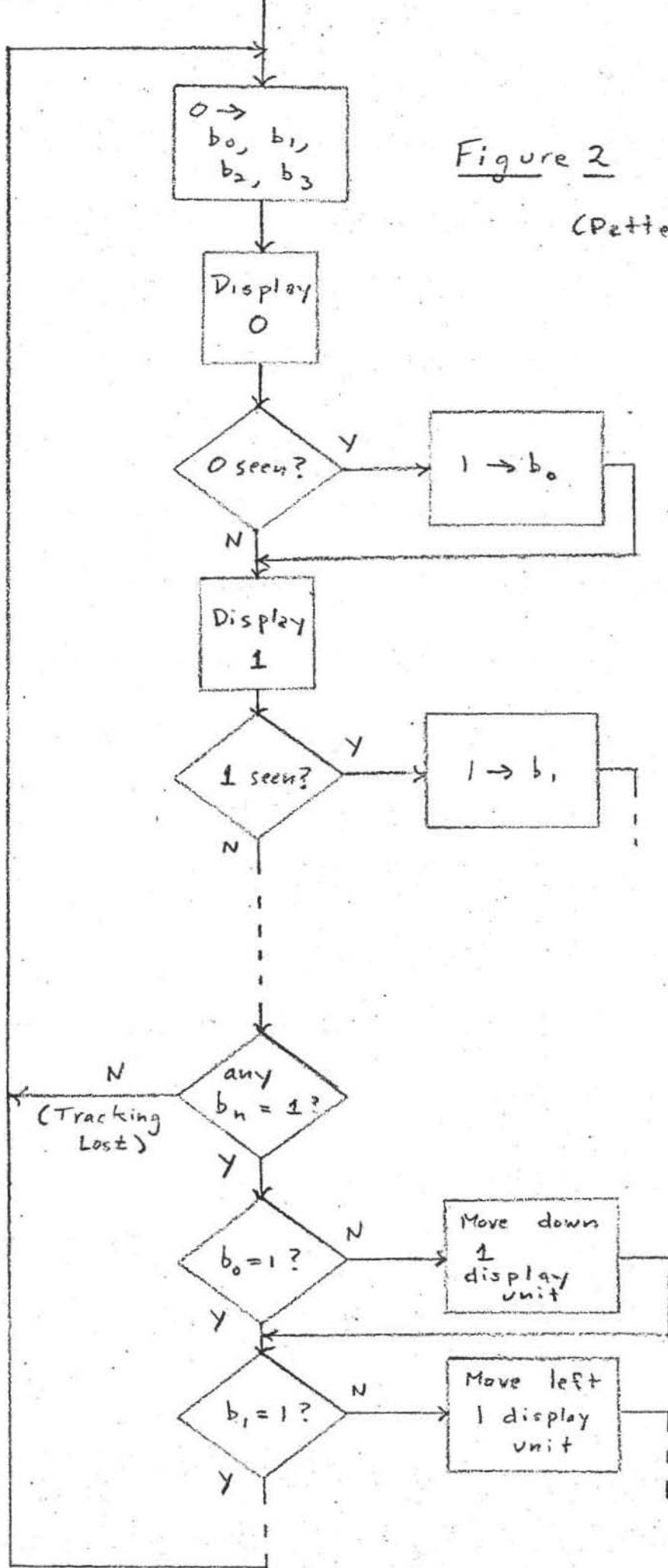
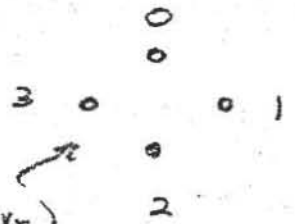
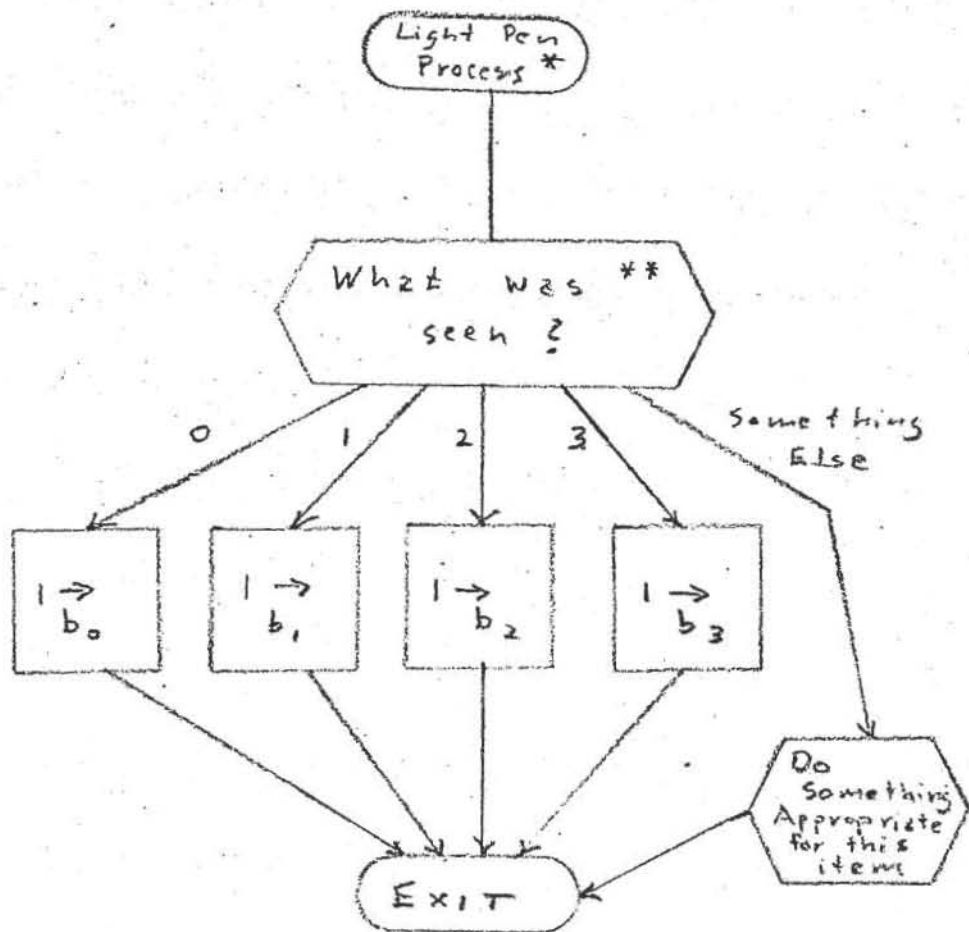
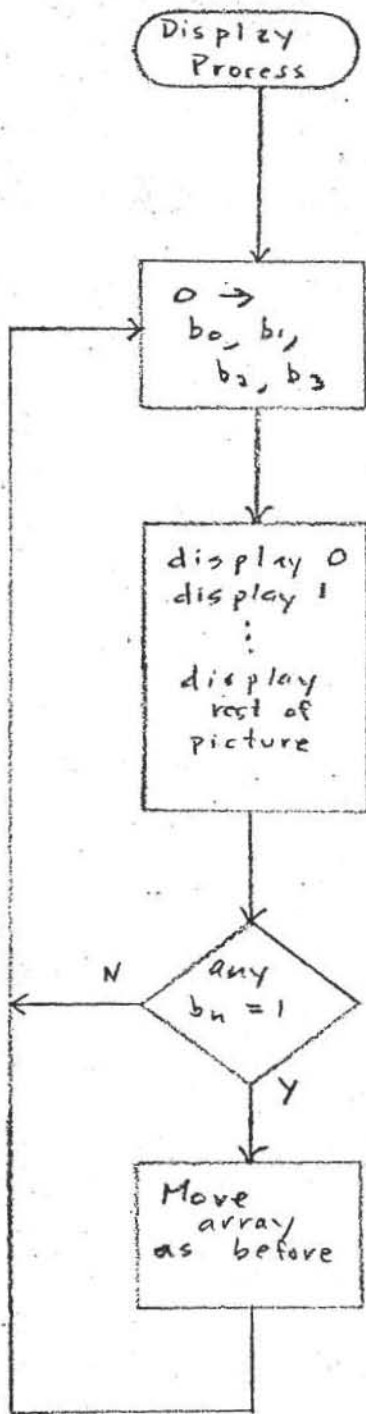


Figure 2

(Pattern)





* Started when pen sees light.

** Look at display address,
what is being displayed

Figure 3

If light from the cathode ray tube is the only clue as to where the light pen is, how is it possible to move the tracking pattern across background information without interference? Obviously the background information and the tracking pattern itself will be located in separate places in the display file table. The codes which are stored to indicate which parts of the display drawing are seen will distinguish between the parts of the tracking pattern and the parts of the background view. Thus the processor which examines the information seen during a complete frame of the display can distinguish between parts of the tracking cross and parts of the background view.

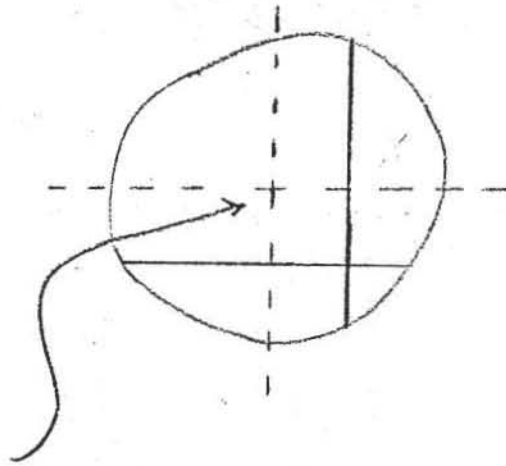
Types of Tracking Crosses

Various types of tracking crosses have been used to detect the position of the light pen. The simple four-dot tracking square described above has the difficulty that if the tracking square is significantly smaller than the field of view of the light pen, then the tracking square would not respond to small motions of the pen. In effect, the tracking cross is a sort of pea underneath a cup formed by the field of view of the light pen such that the pea moves only when the wall of the cup actually touches it. On the other hand, if the four dot tracking square is made larger so that it fills the entire field of view of the light pen, there is a risk that the field of view of the light pen may become smaller than the tracking square. If the pen field of view is smaller than the tracking square, it may slip inside the tracking square and tracking will be lost.

The solution to this dilemma is to make the size of the tracking square adjustable to match the size of the field of view of the light pen. In effect, the tracking square should feel out the edges of the light-pen field of view in four directions, and compute the center of the light pen field of view from the information thus derived.

On a point-plotting scope, the usual technique for making a pen-tracking cross is to display four points immediately adjacent to the estimated pen location, and if they are seen, four points adjacent to them but further out in both directions along both axes. When a point in some arm is no longer in the field of view of the pen, we know the coordinates of that edge of the light-pen field of view. Averaging the vertical positions of the two points on the vertical axis gives the vertical coordinate of the pen center, and similarly the horizontal coordinate is the average of the two points on the horizontal axis. The process of scanning out from the center gives the display the appearance of a cross, hence the name, "tracking cross", for this type of tracking. So far as I know, such a precision tracking cross was first used by T. G. Stockham in April 1959 on the TX-0 computer at MIT.

If the display contains a line-generator, it is more sensible to start drawing lines inward from four points outside the predicted field of view, measuring the edge of the field when the light-pen interrupt logic announces a hit - i.e. when the line has just entered the sensitive region. The appearance of these displays is shown in Figure 4.



MEASURED
PEN POSITION

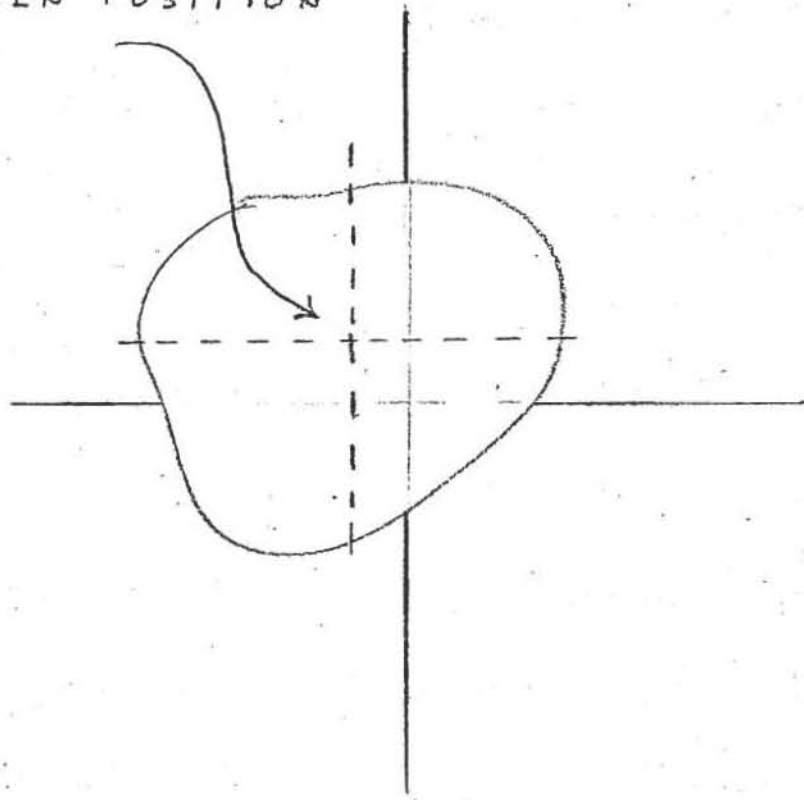


FIGURE 4 - TRACKING CROSSES

Many other forms of tracking squares can be devised to do this job. The only important characteristic that they must have (and the rest of this appendix is to show that that's so) is that they must determine the edge of the field of view of the tracking cross with minimum noise. Any noise present should preferably be as smooth as possible. Consistent errors in the measurement of position are tolerable, but inconsistent errors with high frequency components (noise) are not. The reason that a low noise measurement of the pen position is important is that the pen position predictor serves to amplify the measurement noise. If a lower noise measurement can be made, a higher order predictor can be used, and tracking can be done less often.

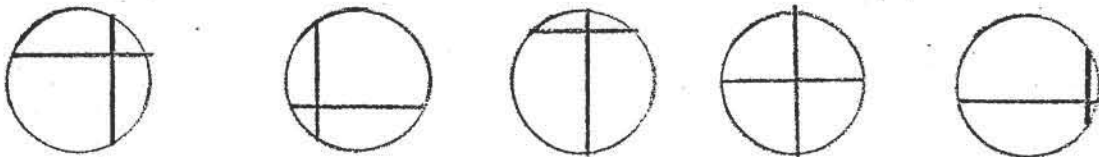
Low noise pen position measurement is achieved by several techniques. First, it is important to measure pen position to the finest resolution available on the scope. The quantization error of position will show up directly as a noise in the measurement, where

$$|\text{noise}| = \frac{1}{\text{resolution}}$$

Second, a linear search procedure should be used. The successive points displayed should reach the edge of the pen in linear sequence either from inside or outside. If a logarithmic search is used, motions of the pen during sampling time will introduce errors larger than the resolution used. Third, the measurements for the four edges of the field of view should be made as nearly simultaneously as possible. If the measurements are made at different times and the pen is moving, the pen's field of view will effectively be distorted in shape.

A Common Misconception

It is a common misconception that the pen-tracking program is a servomechanism in the sense that it is a feedback device. This is just not so. In fact, the pen-tracking program is a measurement system wherein the position of the center of the field of view of the light pen is measured by the tracking cross. Any tracking cross capable of measuring the center of the field of view of the light pen will obtain the same measurement, provided only that the field of view is circular. The several crosses shown below, for example, all agree on the position of the pen field of view.



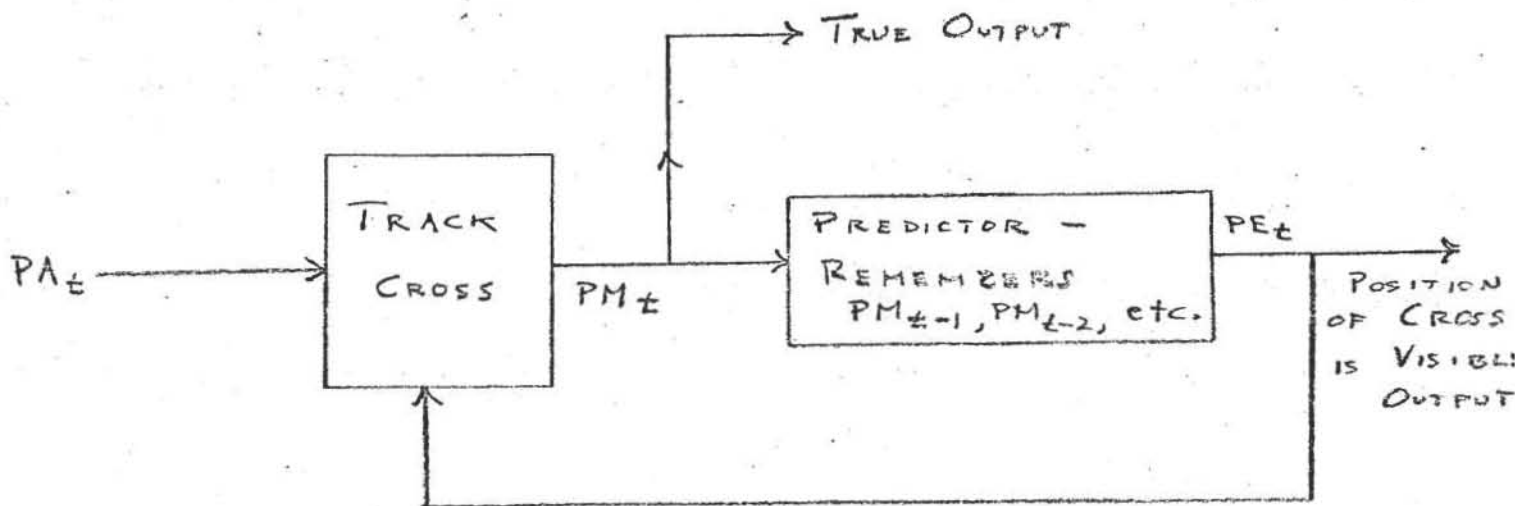
In order to actually make the measurement of the pen location, the tracking program needs an estimated pen position about which to draw the tracking figure. The particular value of the estimate makes no difference whatsoever, provided only that it lies within the pen field of view. i.e. so long as

$$|\text{actual position} - \text{estimated position}| < \text{radius}$$

otherwise tracking will be lost. The rest of this appendix concerns itself with methods of arriving at a suitable estimate. Let me remind you again that tracking is not a feedback system because the measured pen location is not a function of the estimated pen location.

Pen-Position Prediction

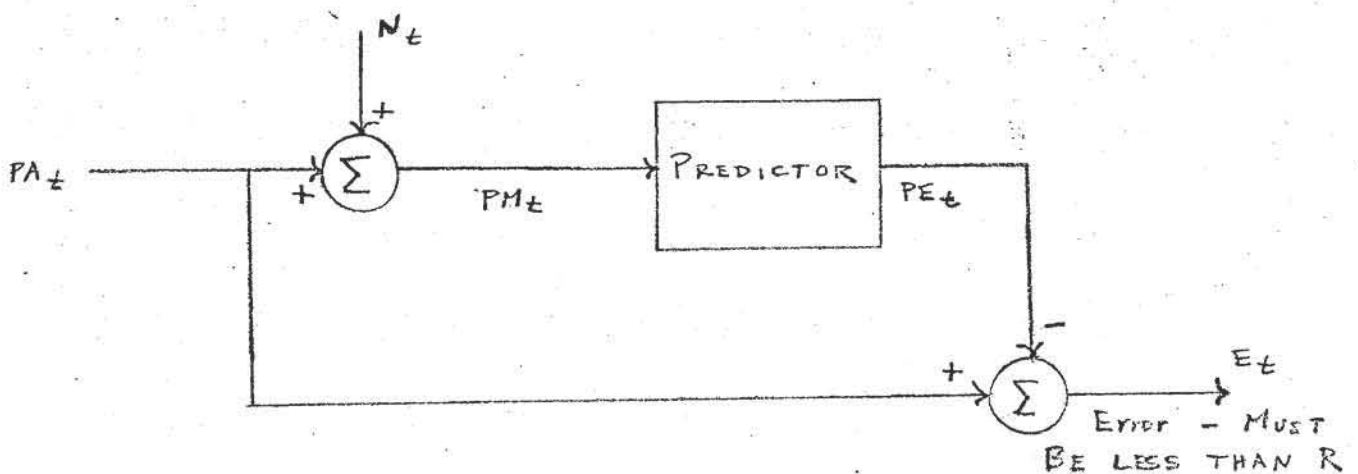
A complete pen-tracking system will use some kind of prediction scheme for arriving at the pen-position estimate to be used for each tracking cross. If we call the actual pen position "PA", the measured pen position "PM", and the estimated pen position "PE", and use subscripts "t", "t-1", etc. to indicate the sampling time at which these numbers are valid, then a block diagram of the pen-tracking process looks like this:



The estimated position at time "t" is used only to establish a tracking cross within the light pen field of view. The particular value of the estimated pen position does not effect the position

measured at that time. What you observe on the scope, however, is the tracking cross whose center is the estimated pen position. Perfectly stable tracking is possible even if the estimated pen position changes radically throughout the pen field of view. In other words, the mere observation that the pen-tracking cross is "jumping around" within the pen field of view does not mean bad tracking, because the measured pen position may nevertheless be quite stable.

In order to see more clearly what the effect of noise on the predictor is, let us redraw the block diagram of the system as follows:



It is now evident that if PA is constant, the measurement noise signal is passed into the predictor box, and may be amplified by the predictor box to produce a very noisy pen position estimate. Design of the pen position estimating box must insure that the magnitude of the noise as amplified by the pen-position estimator does not exceed the radius of the light pen.

Obviously the pen-position estimator can work only on past data. That is, the estimate of the pen position $PE_t = F(PM_{t-1}, PM_{t-2}, \dots)$. If we choose to use linear prediction where N is called the order of the prediction

$$PE_t = \sum_{i=1}^n a_i PM_{t-i}$$

The error signal, however, will be

$$\begin{aligned} E_t &= PA_t - PE_t = -N_t + PM_t + PE_t \\ &= -N_t - \sum_{i=0}^n a_i PM_{t-i} \end{aligned}$$

where the coefficient $a_0 = -1$.

Suppose that we want to predict in such a way that pen motions with N constant derivatives are predicted with zero error. For such motions, the N^{th} differences in pen position will be constant. If we define a delay operator, D , such that,

$$D(P_t) = P_{t-1}$$

then the difference operator is $[1-D]$, and

$$[1-D](P_t) = P_t - D(P_t) = P_t - P_{t-1}$$

The N^{th} difference operator is $[1-D]^N$. Thus the appropriate coefficients a_i to use are given by

$$E = [1-D]^N P(t)$$

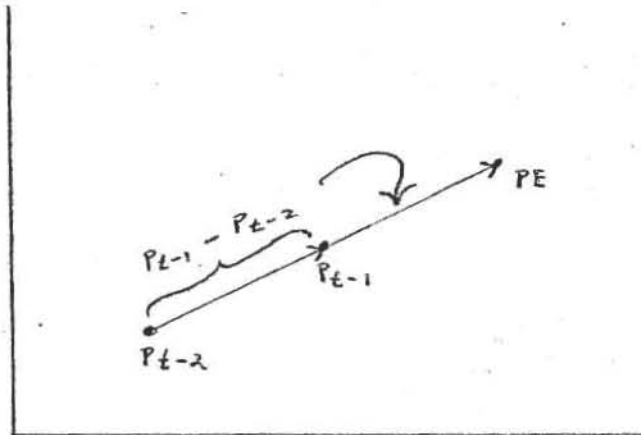
The appropriate values for the coefficients are:

N	a_0	a_1	a_2	a_3	a_4	
1	-1	1				Constant Position Assumption
2	-1	2	-1			Constant Velocity Assumption
3	-1	3	-3	1		Constant Acceleration Assumption
4	-1	4	-6	4	-1	etc.
5	-1	5	-10	10	-5	

Where N is one, the estimate is just the previous position which is correct only if the pen is not moving. If N is 2, the estimate is

$$PE = P_{t-1} + (P_{t-1} - P_{t-2})$$


which is correct assuming that the velocity is constant.



In actual practice, it is practical to use pen-position predictors up to order 3 (constant acceleration), but as we shall see, predictors of higher order amplify noise too much to be of practical value.

Suppose that the successive noise signals during the measurements happen to come out in such a way as to make the worst possible prediction. How would that be? It is clear from the signs of the coefficients in the table, that if the noise errors alternated in signs, the error would be of maximum magnitude. In other words when the noise frequency is just one half of the sampling frequency. Under such conditions, the noise signal will cause an error which is larger than itself. In fact, the noise signal as amplified would be:

$$\text{Noise Error}_t = \sum_{i=0}^N a_i N_{t-i} = |N| \sum_{i=0}^N a_i (-1)^i$$


 AMPLITUDE OF NOISE

which means that predictors amplify noise by the factors shown below.

N	a ₀	a ₁	a ₂	a ₃	a ₄	<u>Noise Gain</u>	
1	-1	1				2	Constant Position Assumption
2	-1	2	-1			4	Constant Velocity Assumption
3	-1	3	-3	1		8	Constant Acceleration Assumption
4	-1	4	-6	4	-1	16	etc.
- etc. -							

Since the maximum amplitude of the noise signal is roughly \pm one scope unit, a constant acceleration predictor will cause the tracking cross to bounce around about \pm 8 scope units. Since light pens typically have a field of view of about 1/2 inch diameter, which is about 50 scope units in diameter or 25 scope units in radius, you would expect to be able to use prediction of order 4. It is easy to see, however, that marginal operation might well result.

Rotating Light Pen Tracking

It would be desirable to be able to input information to the machine about the orientation of the picture presented. One way to input rotational information is to specify with a pointer the two endpoints of some line in the picture. This would yield both translational and rotational information. We can consider, however, the possibility of discovering two pairs of coordinates simultaneously, as with a device like that of Figure 5. The two pen styli have independent "read" hardware, or perhaps share a common processor, but yield separate coordinates to the computer. The average position

$$\bar{x} = \frac{x_1 + x_2}{2} \qquad \bar{y} = \frac{y_1 + y_2}{2}$$

and the angle of inclination of the line determined by the two pens

$$\tan \alpha = \frac{y_2 - y_1}{x_2 - x_1}$$

(to within an ambiguity of 180°) can be computed to give orientation and position information.

At first, it might appear that this is about the only way of simultaneously reading a position and a direction from a standard input device. If, however, a lightpen is fitted with a non-circular aperture, there is a method for duplicating the performance of the pen device described above. The position of the pen can be tracked, and measurements of its orientation can also be made.

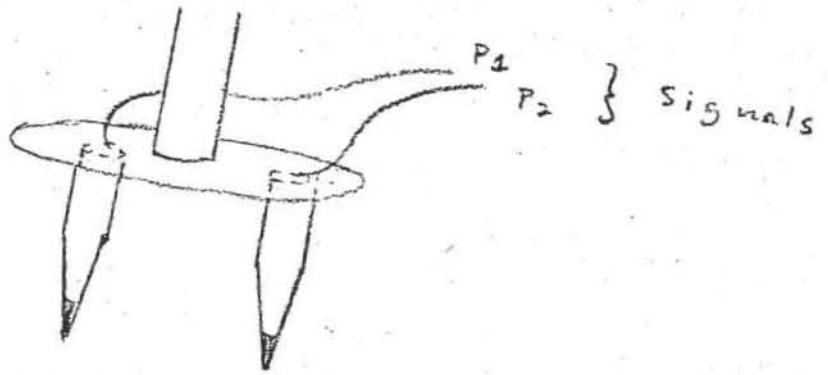


FIGURE 5

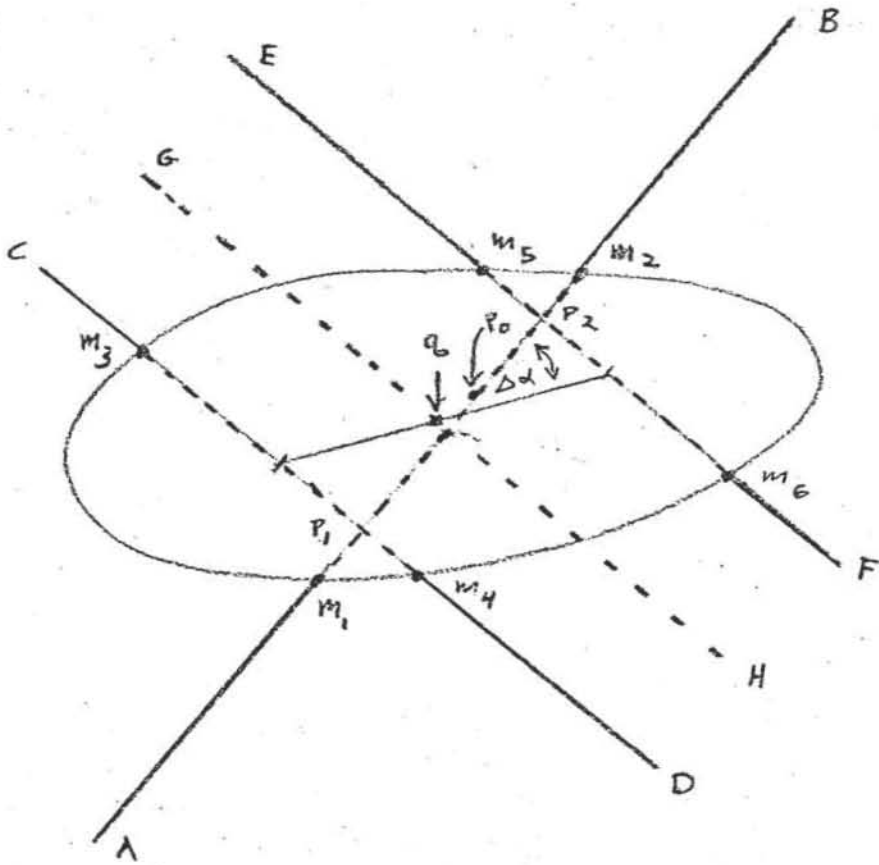


FIGURE 6

Consider the "tracking cross" of Figure 6. Superimposed on it is a hypothetical lightpen window. The window shape itself is a function not only of the aperture size and shape but also of the distance of the pen from the tube and the azimuthal angle at which the pen is held. In practice, one makes a guess on the lightpen position, say P_0 , and the angle of inclination of the window, say α_0 . The components of the cross are then drawn. First, the lines thought to be aligned with window orientation are drawn from the points

$$[P_{0x} + d\cos\alpha_0, P_{0y} + d\sin\alpha_0]$$

$$[P_{0x} - d\cos\alpha_0, P_{0y} - d\sin\alpha_0]$$

toward the point P_0 . In the process, the points m_1 and m_2 are measured as the lightpen "sees" the first point displayed within the window. We now know something about the size of the window (the "image" of the lightpen aperture on the screen). If the lines \overline{CD} and \overline{EF} are now displayed such that they intersect the line \overline{AB} somewhere within the area bounded by m_1 and m_2 , we are virtually assured of a hit. In practice, it works to choose the point P_1 as

$$\frac{3m_2 + m_1}{4}$$

and P_2 as

$$\frac{3m_1 + m_2}{4}$$

The lines \overline{CD} , \overline{EF} are then displayed and hits recorded. We now have six measurements, $m_1 \dots m_6$. We suspect that the position of the component of the new pen position q along the line AB should be determined by the points m_1 and m_2 , since the window intersects AB at nearly right angles where these

measurements are made. Similarly, we would like to determine the new position by using the measurements $m_3 \dots m_6$.

Consider the average points P_3 and P_4 where

$$P_3 = \frac{m_3 + m_4}{2}$$

$$P_4 = \frac{m_5 + m_6}{2}$$

Since the lines \overline{CD} and \overline{EF} were chosen perpendicular to \overline{AB} and such that $\overline{P_2 m_2} = \overline{P_1 m_1}$, then the average of P_3 and P_4 is guaranteed to be on line \overline{GH} . (\overline{GH} is the locus of points equidistant from m_1 and m_2 .) (See Figure 6)

We can compute

$$q_x = \frac{m_{3x} + m_{4x} + m_{5x} + m_{6x}}{4}$$

$$q_y = \frac{m_{3y} + m_{4y} + m_{5y} + m_{6y}}{4}$$

and

$$\tan \alpha = \frac{P_{4y} - P_{3y}}{P_{4x} - P_{3x}} = \frac{m_{5y} + m_{6y} - m_{3y} - m_{4y}}{m_{5x} + m_{6x} - m_{3x} - m_{4x}}$$

Various error-correction schemes can be devised. If the lightpen sees line \overline{AB} but not \overline{CD} , then x is incremented by 90° and the process starts over. Note that in practice, only

$$\Delta x = \beta(m_{5x} + m_{6x} - m_{3x} - m_{4x})$$

$$\Delta y = \beta(m_{5y} + m_{6y} - m_{3y} - m_{4y})$$

are stored, which obviates looking up tangents. β is a normalization constant, chosen so that $(\Delta x)^2 + (\Delta y)^2 = \text{some constant (say } 2500_{10})$.

At first glance, there would appear to be stability problems. Consider the window and cross in Figure 7. Clearly m_1 and m_2 are rather "fuzzy" figures. Therefore, the position may be in error. Using fairly long lines ($\sim 2''$), the error scheme described above seems to avoid gross instability.

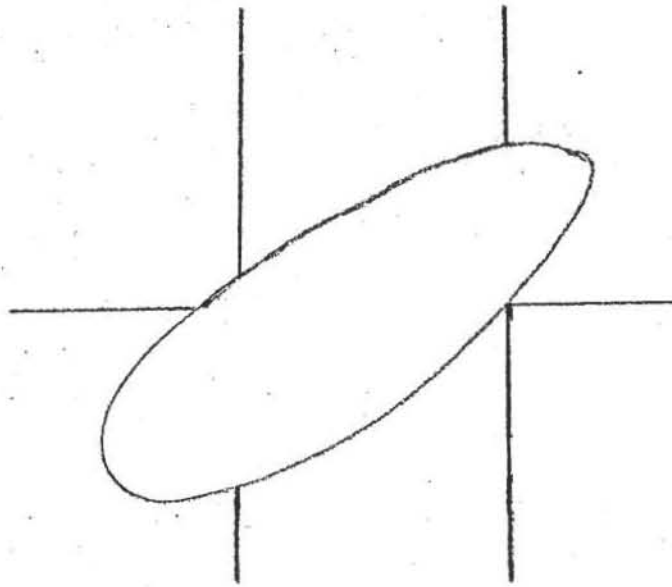


FIGURE 7