PC Stereographics

by

Robert J. Sciamanda
3110 West 40th St
Erie, PA 16506

Introduction

According to the Encyclopedia Brittanica "Stereography is uniquely photographic, since no artist could draw two scenes in exact perspective from viewpoints separated only 2 1/2 inches (six centimetres) - the normal distance between human eyes". Well, what may be impossible for the artist is really a simple task for a computer, even a PC running BASIC!

It is an elementary exercise in projective geometry to develop the algorithm for generating stereograms, which can be viewed either directly on a monitor screen or in hard copy from a printer. The technique which we will develop in this article will be useful both for the practical display of computer generated graphics in stereoscopic 3D, and as an experimental tool for investigating the psycho-physical mysteries of three dimensional vision itself. Along the way we will learn how to produce perspective graphics, since a stereogram is simply a pair of perspective views of the same scene.

How We Have Learned To See In 3-D

Figure 1 illustrates the geometry involved as each eye lens forms images of external object points on the retina. Actually, the eye lens converges a bundle of light rays from each object point onto the corresponding retinal image point; in the interest of clarity, only the "chief" ray representing each bundle is shown here. Because of the eye separation, the left and right retinal images are not identical. In the two views of a three dimensional scene, the relative separations and orientations of object points will be different. Our brains have learned to "fuse" these disparate images into one, interpreting the disparities as the result of three dimensional separations of object points. Thus, remarkably, we see three dimensional depth in a pair of two dimensional retinal images.

Simulation By Projections

In Figure 1, the eyes are shown looking through a window; this window is our "projection plane". From the perspective of the left eye, A' and B' are the projections (onto the window) of the object points A and B. If, instead of looking at the object points A and B, the left eye were presented with simulated object points at A' and B', the chief ray paths through the eye lens would be identical and the retinal point image positions would be identical. Thus, for each eye, the projections (onto a two dimensional plane) of a set of object points can simulate those object points to the eye, producing identical retinal projections.

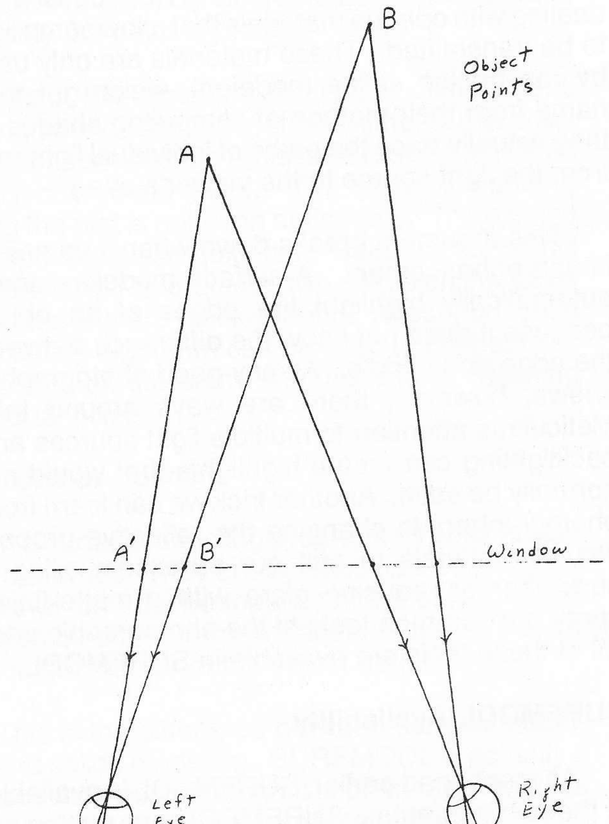

Figure 1. In binocular vision, a different perspective projects a different image onto each retina. The brain reads three dimensional depth into the disparities between the two flat images. Perspective projections onto an intervening "window" can simulate the object points.

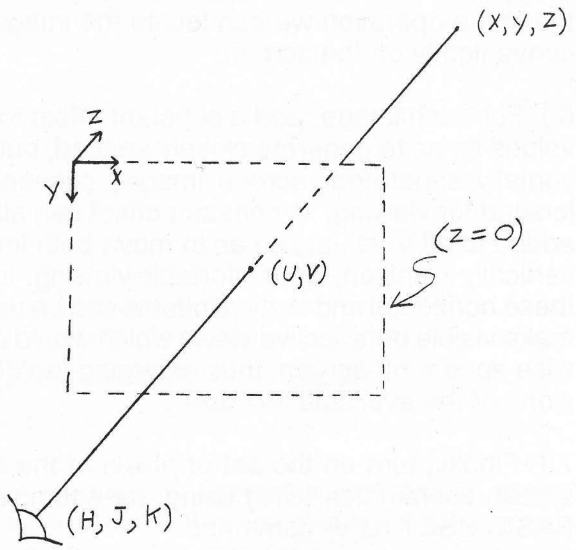

Figure 2. The geometry of the perspective algorithm. Turning on the pixel (U,V) of the monitor screen at (Z=0) will simulate the object point (X,Y,Z) to the eye at (H,J,K); similarly another pixel will simulate the same object point to the second eye.

Our task is clear: we consider that we are viewing a distant object through our monitor screen, which becomes the projection window, and we turn on a screen pixel at the projection of each object point onto the screen. We do this twice, first projecting from our left eye as vertex, and then from our right eye, thus generating a stereographic image pair.

Developing The Algorithm

Referring to Figure 2, we take the upper left corner of the monitor screen as the origin for all coordinate measurements. The X axis is horizontal and positive to the right; positive Y is vertical and down; positive Z is into the monitor. We draw a line joining the eye at (H,J,K) and an object point at (X,Y,Z). This line intersects the screen at (U,V), the pixel which we should turn on to simulate the object point at (X,Y,Z) to the eye at (H,J,K). In order to do this for an entire set of object points, and for two eyes, we need to develop expressions giving U and V as functions of X,Y,Z,H,J and K.

The problem reduces to that of finding the coordinates of the point of intersection of a line and a plane. This is done by solving the equations of the line and the plane as a set of simultaneous equations. The general equation describing a line going through the two points (X1,Y1,Z1) and (X2,Y2,Z2) is:

$$(X-X1)/(X1-X2) = (Y-Y1)/(Y1-Y2) = (Z-Z1)/(Z1-Z2),$$

The equation of the screen plane is simply: $Z = 0$.

Solving these equations simultaneously gives:

$$(X-X1)/(X1-X2) = -Z1/(Z1-Z2) \text{ and}$$
$$(Y-Y1)/(Y1-Y2) = -Z1/(Z1-Z2) .$$

We apply these results to our situation by replacing the coordinates of the intersection point (X,Y) with the screen pixel coordinates (U,V); the point (X1,Y1,Z1) is replaced by the general object point (X,Y,Z), and the point (X2,Y2,Z2) is replaced by the eye location (H,J,K). Converting to this notation and solving the two equations for U and V, respectively, gives:

$$U = X - Z(X-H)/(Z-K) \text{ and}$$
$$V = Y - Z(Y-J)/(Z-K) .$$

This last equation pair is the desired algorithm, giving the screen pixel coordinates (U,V) of the projection of the object point (X,Y,Z) onto the monitor screen plane (Z=0) from the perspective of an eye at (H,J,K).

The Program Flow

In our programming we shall describe all spatial coordinates in inches; just before displaying the results we will convert the screen coordinates (U,V) from inches to screen pixels. The program flow will be (using the upper left corner of the monitor screen as origin for all coordinates):

1.) Generate numerical values for the coordinates (X,Y,Z) of the set of object points as they would exist in real space behind the monitor screen at around Z = 20 inches or more. Usually, this will be the most difficult and painstaking task. In the accompanying program "MAIN", a different technique of object data construction is used in the construction of each of the five scenes depicted; let these serve as examples for your experimentation.

Do not be over ambitious in detailing the object scene. Begin by using only a sparse sampling of object points and lines, sufficient to describe the outlines and geometric features of the object scene. Without gray-scale capability we can only implement a "wire frame" or line drawing representation, not a photographic rendition. To facilitate fusion of the resultant images, the horizontal extent of the sampled object scene must be restricted so that its projection on the window/screen (at Z=0) will not exceed 2 inches, as viewed from an eye located 10 inches in front of the screen. The only limitation in the vertical direction is the screen itself.

2.) Define the two eye locations (H,J,K) to be located in front of the screen at about K = -10 inches, for normal focusing of the eye lenses on the screen image. Assign a value to J (the same for both eyes) so as to give the desired vertical perspective view of the object as seen through the screen "window".

3.) In the same way assign numerical values to the eye H coordinates so as to give the desired horizontal perspective view. However the value of H for the right eye must be 2.33 inches (6 cm.) larger than H for the left eye.

4.) Using the above developed perspective algorithm, transform the set of (X,Y,Z) object points into the set of projected screen points (U,V). Do this twice, first using the left eye value of H, then using the right eye H value, thus generating two screen images, one for each eye.

5.) Convert the set of (U,V) values, now in inches, into horizontal and vertical screen pixel coordinates. In the PC "Hi- Res" mode, U can range from 0 to 639 pixels; V ranges from 0 to 199 pixels. A self-explanatory program "CAL" is included to generate the necessary scaling factors for your monitor. The program "MAIN" already contains (statement 50) default scale factors for a 13 inch monitor.

The above procedure will usually result in left and right images which overlap on the screen. (This complication was avoided in Figure 1 by exaggerating the eye separation relative to other dimensions.) With sufficiently high display resolution, such overlapping images can be viewed stereoscopically through some mechanism which restricts the view of each eye to its corresponding image (eg. using polarized light, color encoding, and/or time sequencing). Since our stereogram viewing technique requires separated images, we will usually have to add an artificial separation; in the same operation we can locate the image pair conveniently on the screen:

6.) For each image, add a constant offset to all U values so as to generate closely spaced, but horizontally separated screen images conveniently located for viewing. A constant offset can also be added to all V values, so as to move both images vertically in unison, for comfortable viewing. In fact, these horizontal and vertical offsets can be used to make visible perspective views which would otherwise appear off-screen, thus enlarging the dimensions of the available "window".

7.) Finally, turn on the set of pixels at the set of screen coordinates (U,V) using, for example, the BASIC PSET (U,V) command.

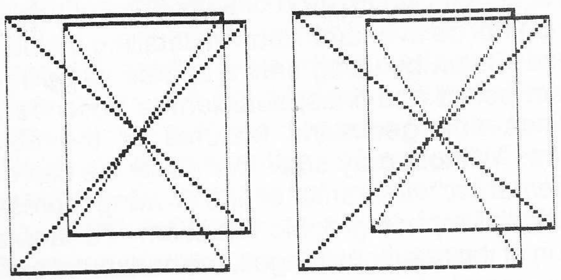

Figure 3. "Unfinished Cube with Diagonals". Stereogram generated by option 1 (statements 215-430) of the program "MAIN". Object points are read from DATA statements, transformed to screen coordinates, and joined with lines. As with the other listings, the perspective can be changed by user input. See text for viewing procedures.

Viewing Stereograms

Point your index fingers upward, holding them about 10 inches in front of your eyes and about 4 inches apart. Gaze between your fingers at a distant object; you will see four finger images. As you slowly bring your fingers closer together, the two inside images will overlap when your fingers are about 2 inches apart. Under these conditions, the left eye's retinal image of the left finger is (mentally) superimposed upon the right eye's retinal image of the right finger. If your brain had reason to believe that these two superimposed images were a stereo pair of views of a single object, it would attempt to fuse them into a 3-D interpretation of that object.

Now replace the two fingers with a stereographic image pair, such as Figure 3. Again gazing straight ahead at a distant object, hold the page about 10 inches in front of your eyes and slowly raise it into view from below, maintaining your gaze on the distant object, as if the page were a transparent window. As with the fingers, you should see three images; again the middle image is a mental superposition of the left eye's view of the left drawing and the right eye's view of the right drawing. Tilt your head or the page to improve the vertical registration; adjust the convergence of your eyeballs to improve the horizontal registration. Concentrate on the middle image until your brain learns to interpret this fusion of two images into one 3-D scene.

If you normally wear corrective eyeglasses, keep them on; bifocals should be used in the near-vision mode. There are exceptions; many near-sighted persons, including the author, find stereogram fusion much easier without wearing their

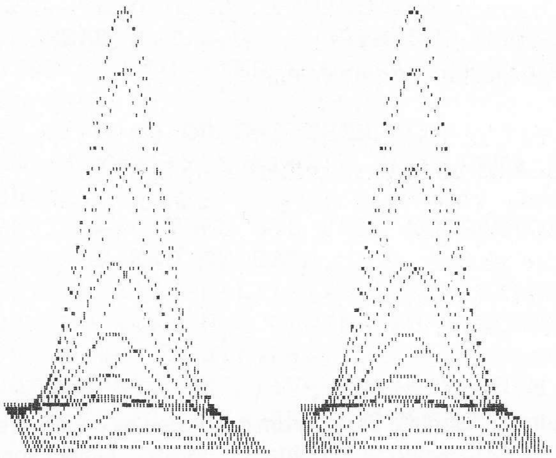

Figure 4. "Mountain". Stereogram generated by option 2 (statements 1215-1350) of the program "MAIN", using a Gaussian function of the form Y=EXP(-X2 - Z2). Object construction proceeds by scanning (X,Z) space and evaluating Y=F(X,Z) at sampled points.

glasses. Experiment for your best viewing technique.

Success requires that you converge your eyeballs toward a distant (fictitious) point, ie. look straight ahead, while focusing your eye lenses upon the nearby page. This is not a natural use of your eyes; normal binocular vision automatically correlates convergence and focusing. With practice and patience, however, most people can become proficient at this stereographic viewing technique.

A pair of convergent lenses can be employed as an aid to restore the correlation between convergence and focusing. As shown in Figure 8, the lens centers are spaced slightly farther apart than the 6 cm. eye spacing so that the eyes use only the inner halves of the lenses, where the lenses behave like shallow prisms. Light rays from an object directly in front of an eye are deviated by the "prism" so that the object appears to be displaced toward the nose. Thus, the eyeballs need no longer maintain a straight ahead gaze toward a distant point; they can now converge toward a nearby point, just as if a nearby solid object were being viewed. (Depending upon their focal length, the lenses may also offer a moderate magnification, but this is incidental to their primary purpose in this application.)

A matched pair of appropriate lenses can often be found as the objectives in a pair of binoculars or opera glasses, even the child's toy variety. The lenses should be at least an inch in diameter; the focal length may be anywhere between 6 and 15 inches. The author can supply a pair of lenses for $10.00 postpaid (Robert Sciamanda, 3110 W. 40 St, Erie, PA 16506).

In general, corrective eyeglasses should be kept on, but do not hesitate to experiment. As shown, a cardboard partition can be placed between the drawings so that each eye sees only its own half of the stereogram. In similar fashion, both of these viewing techniques can be used to view stereograms displayed directly on the monitor screen.

About the Author

Robert J. Sciamanda is a physicist at American Sterilizer Company (AMSCO). Previously he taught physics at Gannon University and designed automated inspection devices for EG&G at the Idaho National Engineering Laboratory. His address is 3110 W 40 St. Erie, PA 16506.▲

Program Listing

Program 1. "CAL". Shows the monitor screen measurements to be made for determining screen scale factors to be inserted in statement 50 of "MAIN". As printed, "MAIN" already contains scale factors appropriate for a typical 13 inch monitor screen.

```
10 REM "CAL" ' MEASURES DISPLAY AREA
PC STEREOGRAPHICS/SCIAMANDA
20 KEY OFF:CLS:SCREEN 2:LOCATE 15,10
30 PRINT "SX = WIDTH OF THIS REC-
TANGLE IN INCHES":PRINT:PRINT
40 PRINT TAB(10);"SY = HEIGHT OF THIS
RECTANGLE IN INCHES":PRINT:PRINT
50 PRINT TAB(10);"INSERT THESE VALUES
IN LINE 50 OF
";CHR$(34);"MAIN";CHR$(34)
55 PRINT:PRINT:PRINT TAB(15);"HIT ANY
KEY TO EXIT";
60 PSET (0,0):DRAW "R639D199L639U199"
70 IF INKEY$="" THEN 70
80 CLS
```

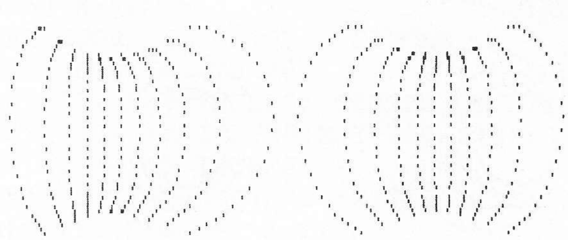

Figure 5. "Saddle". Stereogram generated by option 3 (statements 2215-2350) of the program "MAIN", using the functional form Z = Y2 -X2 . Object construction proceeds by scanning (X,Y) space and evaluating Z=F(X,Y) at sampled points.

Program 2. "MAIN". Written in Microsoft BASICA for the IBM PC and compatibles. Uses high resolution graphics to display a stereogram pair for each of five different scenes. A different technique is illustrated in the construction of each of the five scenes.

```
10 REM "MAIN" ' PC STEREOGRAPHICS/
SCIAMANDA
20 KEY OFF:CLS
30 DIM H(2),OFX(2)
40 PX=640:PY=200:              :REM SCREEN
DIMENSIONS IN PIXELS
50 SX=10: SY=6.5               :REM SCREEN
DIMENSIONS IN INCHES - FROM PROGRAM
"CAL"
60 DX=PX/SX:DY=PY/SY           :REM DISPLAY
SCALE FACTORS (PIXELS/INCH)
61 PRINT"                    SCENE MENU
(Enter a single number)"
62 PRINT
63 PRINT"1 CUBE          2 GAUSSIAN        3
SADDLE      4 TUNNEL        5 LIGHTS"
64 S$=INKEY$:IF S$=""THEN 64
65 S=VAL(S$):IF S<1 OR S>5 THEN
CLS:GOTO 61
66 ON S GOSUB 220,1220,2220,3220,4220
70 H(2)=H(1)+2.33   :REM GET DEFAULT
PARAMETER VALUES
80 CLS:PRINT "ENTER 1 OR 2 :"
90 PRINT "                  1  AUTO-DEMO
MODE"
100 PRINT "                 2  USER
INPUT MODE"
110 N$=INKEY$:IF N$=""THEN 110
120 N=VAL(N$):IF N<1 OR N>2 THEN 80
125 IF N=1 THEN 210
130 CLS: PRINT "ENTER COORDINATES OF
LEFT EYE { X,Y,Z }"
140 PRINT "IN INCHES FROM SCREEN ORI-
GIN (UPPER LEFT CORNER)"
150 PRINT"POSITIVE X IS TO THE RIGHT,
POS. Y IS DOWN, POS. Z IS INTO THE
```

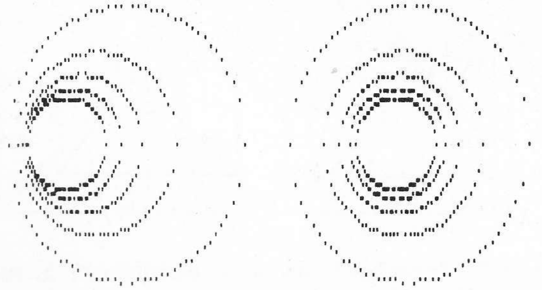

Figure 6. "TUNNEL". Stereogram generated by option 4 (statements 3215-3410) of the program "MAIN". Object data is five coaxial circles of constant radius, displaced in the Z direction.

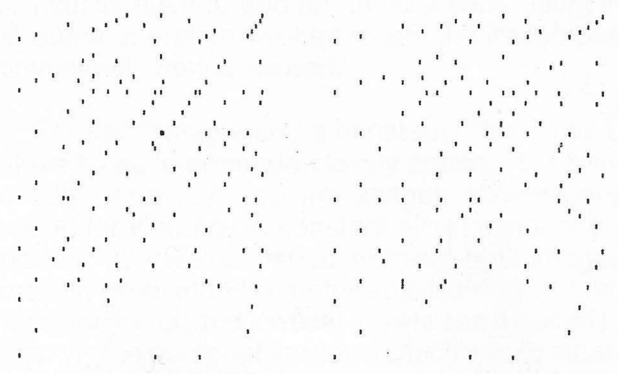

Figure 7. "Lights". Stereogram generated by option 5 (statements 4215-4360) of the program "MAIN". Object space is scanned in all three (X,Y,Z) directions, turning on lights at randomly selected grid points.

```
SCREEN"
160 PRINT: PRINT"DEMO VALUES ARE
";H(1);",";J;",";K
170 INPUT H(1),J,K: H(2) = H(1) +
2.33:PRINT:PRINT:PRINT
180 PRINT"ENTER DISPLAY OFFSETS {
X(LEFT),X(RIGHT),Y } IN SCREEN PIX-
ELS"
190 PRINT: PRINT"DEMO VALUES ARE
";OFX(1);",";OFX(2);",";OFY
200 INPUT OFX(1),OFX(2),OFY
210 CLS:SCREEN 2:ON S GOTO
240,1240,2240,3240,4240
215 REM SCENE 1: "UNFINISHED CUBE"
220 OFX(1)=110:OFX(2)=180:OFY=-30
:REM DISPLAY OFFSETS IN PIXELS
230 H(1)=-1:J=6:K=-10:RETURN :REM
EYE COORDINATES IN INCHES
240 FOR L=1 TO 2          :REM FOR
```

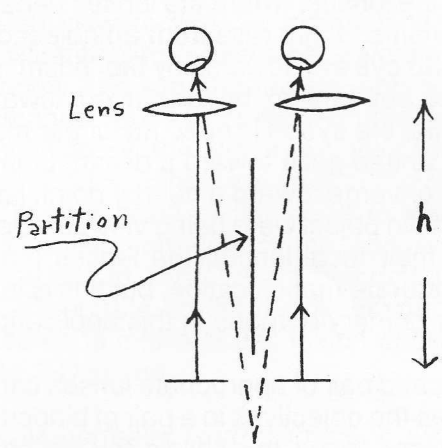

Figure 8. A pair of convergent lenses, used as prisms, can be employed as an aid to viewing stereograms. Adjust the distance "h" for sharpest focus, then vary lens spacing for best fusion. Experiment for your best technique.

```
LEFT, THEN RIGHT PROJECTION
250 READ X,Y,Z,G     :REM GET OBJECT
POINT COORDINATES & OPERATION CODE
(G)
260 IF G =2 OR G=3 THEN
X=X1+X:Y=Y1+Y:Z=Z1+Z    :REM DATA IS
RELATIVE
270 IF G=0 THEN 370 :REM OPERATION
CODE FOR END OF DATA
280 U=X-Z*(X-H(L))/(Z-K):REM TRANS-
FORM TO SCREEN COORDINATES (INCHES)
290 V=Y-Z*(Y-J)/(Z-K)
300 U=U*DX+OFX(L):V=V*DY+OFY :REM
SCALE TO PIXELS; ADD OFFSETS
310 ON G GOTO 330,340,350  :REM
INTERPRETS OPERATION CODE (G)
320 PRINT "DATA STREAM NOT TERMINATED
PROPERLY (END WITH FOUR ZEROES)":END
330 PSET (U,V):GOTO 360        :REM
TURN ON PIXEL (G=1)
340 PRESET (U,V):GOTO 360      :REM
MOVE WITH PEN UP (G=2)
350 LINE -(U,V):GOTO 360       :REM
DRAW LINE (G=3)
360 X1=X:Y1=Y:Z1=Z:GOTO 250   :REM
NEED FOR RELATIVE DATA (G=2 OR 3)
370 IF L=1 THEN RESTORE
380 NEXT L:END
400 DATA 5,1,20,1,6,0,0,3,0,6,0,3,-
6,0,0,3,0,-6,0,3
410 DATA 0,0,6,2,6,0,0,3,0,6,0,3,-
6,0,0,3,0,-6,0,3
420 DATA 6,6,-6,3,0,0,6,2,-6,-6,-
6,3,0,6,0,2,6,-6,6,3
430 DATA 0,0,-6,2,-6,6,6,3,0,0,0,0
1215 REM SCENE 2: "GAUSSIAN MOUNTAIN"
1220 OFX(1)=110:OFX(2)=180:OFY=60
:REM DISPLAY OFFSETS IN PIXELS
1230 H(1)=-1:J=1:K=-10:RETURN   :REM
EYE COORDINATES IN INCHES
1240 DEF FNF1(X,Z)=6-10*EXP(-((X-
5)^2+(Z-20)^2)/2):REM DEFINE Y=F(X,Z)
IN INCHES
1250 FOR X=2 TO 8 STEP .1 :REM SCAN X
& Z (INCHES) OF OBJECT SPACE
1260 FOR Z=17 TO 23 STEP .4
1270 Y=FNF1(X,Z)               :REM
EVALUATE Y=F(X,Z) AT EACH OBJECT
POINT
1280 FOR L=1 TO 2              :REM FOR
LEFT, THEN RIGHT PROJECTION ONTO
SCREEN
1290 U=X-Z*(X-H(L))/(Z-K)     :REM X
COORDINATE OF PROJECTED POINT
(INCHES)
1300 V=Y-Z*(Y-J)/(Z-K)        :REM Y
COORDINATE OF PROJECTED POINT
(INCHES)
1310 U=U*DX+OFX(L):V=V*DY+OFY :REM
CONVERT TO PIXELS; ADD OFFSETS
1320 PSET (U,V)                   :REM
PLOT THE SCREEN PROJECTION POINT
1330 NEXT L
1340 NEXT Z
1350 NEXT X:END
2215 REM SCENE 3: "SADDLE"
2220 OFX(1)=70 :OFX(2)=110:OFY=-20
:REM DISPLAY OFFSETS IN PIXELS
2230 H(1)=2:J=5:K=-10:RETURN :REM EYE
COORDINATES IN INCHES
2240 DEF FNF1(X,Y)=(Y-4)^2-(X-4)^2+60
:REM DEFINE Z=F(X,Y) - ALL IN INCHES
2250 FOR Y=-2 TO 10 STEP .5
:REM SCAN X & Y (INCHES) OF OBJECT
SPACE
2260 FOR X=-1 TO 9 STEP 1
2270 Z=FNF1(X,Y)                 :REM
EVALUATE Z=F(X,Y) AT EACH OBJECT
POINT
2280 FOR L=1 TO 2                :REM FOR
LEFT, THEN RIGHT PROJECTION ONTO
SCREEN
2290 U=X-Z*(X-H(L))/(Z-K)      :REM X
COORDINATE OF PROJECTED POINT
(INCHES)
2300 V=Y-Z*(Y-J)/(Z-K)         :REM Y
COORDINATE OF PROJECTED POINT
(INCHES)
2310 U=U*DX+OFX(L):V=V*DY+OFY :REM
CONVERT TO PIXELS; ADD OFFSETS
2320 PSET (U,V)                   :REM
PLOT THE SCREEN PROJECTION POINT
2330 NEXT L
2340 NEXT X
2350 NEXT Y:END
3215 REM SCENE 4: "TUNNEL"
3220 OFX(1)=50:OFX(2)=90:OFY=-20:
REM DISPLAY OFFSETS IN INCHES
3230 H(1)=2:J=5:K=-10:RETURN     :REM
EYE COORDINATES IN INCHES
3240 DIM X(80),Y(80)              :REM
CONSTRUCT THE ARRAYS X(I),Y(I), GIV-
ING
3250 R=4:A=5                      :REM
THE COORDINATES OF A CIRCLE
3260 FOR I=1 TO 20                :REM
OF RADIUS 4 INCHES; CENTER OF CIRCLE
3270 X(I)=A+I*R/20:X(I+20)=X(I):REM
IS AT X=Y=5 INCHES.
3280 Y(I)=A+SQR(R^2-(X(I)-
A)^2):Y(I+20)=2*A-Y(I)
3290 X(I+40)=2*A-X(I):Y(I+40)=Y(I)
:REM  EACH CIRCLE WILL BE A TUNNEL
3300
X(I+60)=X(I+40):Y(I+60)=Y(I+20):REM
CROSS-SECTION.
```

```
3310 NEXT I
3320 FOR L=1 TO 2                :REM
LEFT, THEN RIGHT PROJECTIONS.
3330 FOR Z=30 TO 110 STEP 20 :REM
PLACE CIRCLES AT Z=30,50,70,90,110
3340 FOR I=1 TO 80
3350 U=X(I)-Z*(X(I)-H(L))/(Z-K)  :REM
X COORDINATE OF PROJECTED POINT
(INCHES)
3360 V=Y(I)-Z*(Y(I)-J)/(Z-K)     :REM
Y COORDINATE OF PROJECTED POINT
(INCHES)
3370 U=U*DX+OFX(L):V=V*DY+OFY    :REM
CONVERT TO PIXELS; ADD OFFSETS
3380 PSET (U,V)                  :REM
PLOT THE SCREEN PROJECTION POINT
3390 NEXT I
3400 NEXT Z
3410 NEXT L:END
4215 REM SCENE 5: "RANDOM LIGHTS"
4220 OFX(1)=120:OFX(2)=170:OFY=-20:
REM DISPLAY OFFSETS IN PIXELS
4230 H(1)=-1:J=7:K=-10:RETURN  :REM
EYE COORDINATES IN INCHES
4240 PRINT"ENTER ANY NUMBER (SEED FOR
RANDOM NO GENERATOR)"
4250 INPUT SD: RANDOMIZE SD: CLS
4260 FOR Y=-1 TO 5             :REM
SCAN OBJECT SPACE
4270 FOR X=3 TO 8              :REM AND
TURN ON
4280 FOR Z= 20 TO 30 STEP 2 :REM
LIGHTS AT RANDOM
4290 IF RND > .5 THEN 4360    :REM
BRANCH FOR NO LIGHT AT THIS POINT
4300 FOR L=1 TO 2             :REM FOR
LEFT, THEN RIGHT PROJECTION ONTO
SCREEN
4310 U=X-Z*(X-H(L))/(Z-K)     :REM X
COORDINATE OF PROJECTED LIGHT
(INCHES)
4320 V=Y-Z*(Y-J)/(Z-K)        :REM Y
COORDINATE OF PROJECTED LIGHT
(INCHES)
4330 U=U*DX+OFX(L):V=V*DY+OFY :REM
CONVERT TO PIXELS; ADD OFFSETS
4340 PSET (U,V)               :REM
TURN ON THE SCREEN PIXEL
4350 NEXT L
4360 NEXT Z: NEXT X: NEXT Y:END
```

Editor's Note:

Disk with the program is available for $19 from ACCESS, PO Box 12847 Research Triangle Park, NC 27709