UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

STEREO

A computer program for projecting and plotting stereograms

by

Peter German[1] and Carl Johnson[2]

Open-File Report 82-726

This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards and stratigraphic nomenclature.

1. Arco
444 South Flower Street
Rm LWF -32-55
Los Angeles, CA 90017

2. U.S. Geological Survey
California Institute of Technology
Pasadena, California 91125

1983

CONTENTS

INTRODUCTION

STEREO is a computer program that projects and plots stereograms to illustrate three-dimensional relationships. The purpose of this report is to make the program available for general use by explaining the concepts and computational methods used, describing how to use the program, and publishing a listing of the program source code.

The development of STEREO was motivated by the need in seismology to describe the spacial distribution of earthquakes. The use of computer generated stereograms is a concise, efficient, and inexpensive method for representing three dimensional relationships. For the study of spacial seismicity patterns, stereograms provide more freedom than the more commonly used epicenter maps and cross sections.

Stereograms generated by earlier computer programs have already proven useful in seismicity studies. Johnson (1979) made extensive use of stereograms to study three-dimensional distributions of earthquakes in swarms, using a computer program developed by Don Eckelman (unpublished) at Colgate University. Stereograms have also been used by Johnson and Richter (1979) to illustrate the seismicity associated with subduction zones. In both these studies, the program used to generate the stereograms, while creating a view which was qualitatively useful, unfortunately introduced some distortions, making it difficult to correctly judge relative distances and angles in the stereo images.

The program presented in this report corrects the distortion problems of previous programs, increases the versatility of the stereograms which can be generated, and simplifies the use of the program. Several methods used by

Eckelman have been adopted in this program, but the projection method is entirely new, and the subroutines have been totally restructured to allow other programmers to easily design their own application for stereo projections.

The stereograms generated by this program, because the projection method closely simulates binocular vision, are easily fused into one stereo image, providing an undistorted representation of the three-dimensional reationships in the original figure.

DEFINITION OF TERMS

The following terms describe various aspects of stereo views and features of the program, and are used in this report to simplify the discussion of both the mathematical methods and the applications of the program.

VIEW POINT: The location from which an object will be viewed is called the view point, and lies midway between an observer's eyes.

CENTER POINT: The point which an observer is looking directly towards (the point of fixation) is called the center point. It is in the center of the field of view, and generally will be the location of the center of an object, or group of objects, being viewed.

JACK: A three-dimensional symbol generated by the program to illustrate point locations in stereo views is called a "jack"; it consists of three orthogonal line segments intersecting at the point being represented.

CENTRAL LINE OF SIGHT: The vector from the view point to the center point is called the central line of sight.

PROJECTION PLANE: The plane onto which a three-dimensional figure is projected.

PROJECTION POINTS: The hypothetical locations of an observer's eyes on opposite sides of the view point are called the projection points. These are the points from which stereo views are projected onto the projection plane.

COMPUTATIONAL METHODS

Binocular vision creates a three-dimensional image by combining the two-dimensional images seen from each eye's slightly different perspective. By taking advantage of binocular vision, stereo views can accurately reproduce the appearance of three dimensions when the two views present each eye with the correct difference in perspective. Program STEREO generates accurate stereo views by employing a projection which is geometrically identical to binocular vision.

Figure 1 illustrates the geometry of a binocular view of two points, A and B, and their corresponding images on each eye's retina, A' and B'. Each eye sees a distinctly different image, but in a visual process called stereopsis the two images are fused and interpreted as a single three dimensional image, with A closer to the eyes than B. The same image can be reproduced in stereo views, and this is done in the program by projecting the points along each eye's line of sight onto an arbitrary prjection plane (figure 2), so that point A is projected along the right eye's line of sight to A_r, and along the left eye's line of sight to A_l. Point B is projected the same way, and a stereo view is created by plotting separately each eye's different projection of points A and B, first from the right eye's perspective (A_r and B_r), then from the left eye's perspective (A_l and B_l). When the two plots are viewed properly, so that each eye sees only one plot (figure 3), points A and B are perceived to be in their actual three-dimensional relationship.

PROJECTION

Given the locations of a sequence of points that describe a three-dimensional figure, the program generates stereo views by projecting the points from the perspective of each eye onto a plane perpendicular to the central line of sight. The projection method is geometrically equivalent to the parallel projection method described by J. Rule (1941) for use in stereo photography. For easy computation, a cartesian coordinate system is used in the program to specify the locations of the data points, and the x-y plane is used as the projection plane, with the center point at the origin and the view point on the z-axis (figure 4).

Figure 5 illustrates the projection of a data point (x,y,z) along the line of sight from the right projection point (or right eye) onto the x-y plane at (x',y'). The values of x' and y' are derived from the relationship in the figure as follows:

Triangle RED is similar to triangle RBA, so

$$\frac{RE}{DE} = \frac{RB}{AB} \qquad (1)$$

But RE = RB - EB, so AB in equation (1) can be expressed as

$$AB = \frac{(RB)\ (DE)}{RB - EB} \qquad (2)$$

and, making the following substitutions

AB = y'

EB = z

DE = y

RB = VC

equation (2) becomes

$$y' = \frac{y(VC)}{VC - z} \qquad (3)$$

where VC is the distance between the view point and center point. Similarly,

$$x' = \frac{(VC)\ (x - VR)}{VC - z} \qquad (4)$$

where VR is one-half the stereo base length.

Equations (3) and (4) are used by the program in subroutine PROJXN to compute the values of x' and y' for each data point. The points (x',y'), after being multiplied by a scale factor, are plotted to create a stereo view.

ORIENTATION

Users of program STEREO are allowed to project figures from any direction they desire by choosing appropriate locations for the view point and the center point. The projection described in the preceeding section, however, requires that the program use a coordinate system with the center point always at the origin and the view point always on the z-axis, so before any points can be projected, a coordinate system must be set up that fulfills these requirements. The program does this by first inputting the view parameters in a coordinate system convenient for users, and then converting to a coordinate system which allows it to use the simple projection method from a view point on the z-axis as described in the previous section.

Input coordinates can be in either a spherical system for use in describing points in relation to the surface of the earth, or a cartesian system. In either system the view point and center point may be anywhere. If the user chooses the spherical system, the program converts to cartesian coordinates, translates the origin to lie on the center point, and rotates the system around the center point so the z-axis contains the view point. If the user chooses the cartesian input system, the input is directly translated and rotated, without any conversion necessary. The rotations used to move the z-axis onto the view point are shown in figure 6— first a rotation by angle α around the z-axis, and then by angle β around the new x-axis.

Finally, the program assumes, in the case of the spherical coordinate system, that the viewer's feet point toward the center of the earth, so when viewing a figure from the south, the north side will be at the top of the resulting plot. If the same area is viewed from the north, however, the southern side will be uppermost in the plot. When using the cartesian coordinate system, the observer's feet are assumed to point toward the origin, Angle γ in the program refers to a rotation around the z-axis required to keep the observer's feet pointed in the proper direction (see fig. 7).

The values of angles α , β , γ are computed in subroutine SETUP, the translation is performed in subroutine TRANS, and then all the rotations are simultaneously computed in subroutine ROTATE.

STEREO BASE LENGTH

Program STEREO projects stereo views from two projection points, corresponding to the two eyes, which are separated by a distance called the stereo base length. The effect of the stereo base length on stereopsis is a central issue in the generation of stereo views, so it is important to understand how the program computes this parameter. The method used to calculate the stereo base length, like the projection itself, is based on the geometry of binocular vision.

It is the separation between the eyes that gives binocular vision the ability to create accurate three-dimensional images, but when looking at objects at large distances from the eyes, the eyes are so close together compared to the distance to the object that the two separate perspectives become indistinguishable, and stereoptic vision cannot function. In order not to limit the usefulness of the program, however, it is essential to be able to view any figure, however far away, stereoptically. This can only be done by increasing the stereo base with increasing distance to objects being viewed.

The easiest way to vary the stereo base while maintaining a good binocular projection is to constrain the angle of convergence at the center point, the angle between the two lines of sight (see figure 8), to be a constant value which allows for good stereopsis. The stereo base length can then be easily computed from the angle of convergence as follows:

$$\text{Stereo base length} = 2\ (VC)\ \tan\ (C/2) \qquad (5)$$

where VC is the distance between the view point and the center point, and c is the angle of convergence. The program fixes the angle of convergence at four degrees, the value of the angle subtended when looking at an object about one meter from the eyes. This value works well for most stereo views.

The angle of convergence can be constrained to another value if it suits the user's purpose, but too large a value (more than about 15 degrees) forces the eyes to become uncomfortably strained, and too small a value (less than about 0.1 degrees) loses the stereo perspective needed for stereopsis. Constraining the angle of convergence in equation (5) is the default method of computing the stereo base length in STEREO.

Fixing the angle of convergence allows the stereo base to vary, producing good stereoptic perspective regardless of distance to the center point, but it constrains the depth of field that can comfortably be fused into one image (Dudley, 1965). If a fixed angle of convergence is used to calculate the stereo base length, there will always be the same range of stereoptic vision in front of and behind the center point, and stereo views with extreme depth ranges, larger or smaller than the constraint, will not be as useful as they could be.

Dudley (1965) has suggested a general relationship between the stereo base length and the permissable distance between the nearest and farthest planes in a stereo photograph:

$$\text{Stereo base length} = \frac{NF}{50(N-F)} \qquad (6)$$

where N is the distance from the view point to the nearest plane, and F is the distance from the view point to the farthest plane. Unfortunately, the distances to the nearest and farthest planes cannot be calculated in the program, because the first point of a figure is plotted before the last point has been input, a feature desirable for efficiency.

By assuming that the center point is halfway between the nearest and farthest planes, equation 6 can be modified for use in the program, as below:

$$\text{Stereo base length} = \frac{(VC - D/2)\ (VC + D/2)}{50\ D} \qquad (7)$$

where VC is the distance from view point to center point, and D is the depth of field, the distance between the nearest and farthest planes.

To take advantage of this relationship, (7), to optimize the stereo base length, users must input the value of D, the depth of field. This optimization is optional so that the program is still useful if it is inconvenient to find the depth of field.

If D is not input, by default the program fixes the angle of convergence, as described above, and in most cases this produces excellent stereograms. However, figures with unusually small or unusually large depths of field may be more useful and more comfortable to view if the stereo base optimization is used.

SCALE AND FOCAL LENGTH

A three dimensional image will be seen when viewing a pair of plots projected as explained above (see section "projection", page 6) but the image will be undistorted only when the plots are viewed in a way which is geometrically equivalent to the parallel projection used to create them. By introducing a few constraints, the viewing geometry can be controlled by a relationship between the scale of the plot and the distance from the plot to the observer's eyes, herein called the plot focal length.

The constraints, illustrated in figure (9), are; first, that the line joining each eye with the center of its corresponding plot (x' and y' in the figure) is perpendicular to the plane of the plot and, second, that the centers of the plots must be separated by the inter-ocular distance of the observer.

Given these conditions, the stereo image will exactly duplicate the original figure when the viewing angles (figure 9) are equivalent to the projection angles for each point.

More specifically, the original point, P (in figure 9a), will appear identical in distance and direction to the stereo image, P' (figure 9b), when

$$\alpha = \alpha$$

and $$\beta = \beta$$

From similar triangles, it follows from the figure that

$$\frac{x'a'}{xa} = \frac{L'x'}{Lx} \qquad (8)$$

$\frac{x'a'}{x\,a}$ is simply the scale of the plot, LX is equal to the distance from view point to center point, and L'x' we are calling the plot focal length, so equation (8) can be written

$$\text{Scale} = \frac{\text{plot focal length}}{\text{VC}} \tag{9}$$

where VC is the distance from view point to center point. This result may seem intuitively obvious, but this simple derivation has been presented to convince the reader that this method of computing the scale results in an undistorted image.

The value of VC is computed in subroutine SETUP from the locations of the view point and the center point, but the value of the plot focal length must be input by the user, since the focal length depends on what type of optical device will be used to view the stereogram. The scale is computed using equation (9) in subroutine SETUP. In subroutine PROJXN, the projected location of each point (x',y' in equations 3,4, page 7) is multiplied by the scale to compute the final location of each point on the stereo plot.

It can be seen from equation (9) that for a given plot focal length, the scale is controlled by VC, the distance between view point and center point. Moving the view point farther from the center point decreases the scale, and moving the viewpoint closer to the center point increases the scale. Consequently, program users can plot at any desired scale by selecting a view point and a center point which are an appropriate distance apart.

Because it is often inconvenient for users to have to calculate the locations of a view point and a center point required for a given scale, there is an option in the program which allows users to directly input the desired value of the scale. The program then moves the view point by the correct distance along the central line of sight to achieve the desired scale.

The scale applies only to points in the projection plane. Objects closer to the veiw point will be plotted at a proportionaltely larger scale, and objects farther away will be proportionately smaller, as required by geometrical perspective.

PROGRAM LOGIC

An explanation of several features of the program may be helpful to readers trying to decifer the program logic.

The program has been organized to allow an unlimited number of points to be plotted while still using efficient matrix operations to process the data. This is done by using a fixed size buffer in which data points are stored, and when the buffer is full, all the points in the buffer are processed as one matrix. The buffer is then cleared so additional points can be stored and processed. Subroutine PAKRAT fills the buffer, and calls the processing routines when it is full.

Another feature is that the subroutines are organized so they can be used by other programmers wishing to design their own uses for stereograms. By simply calling MOVE3D, DRAW3D, or JACK, a programmer can generate stereograms without reprogramming the projection logic. The function and parameters of the key subroutines other programmers may want to use are explained in the appendix, and the subroutine hierarchy is shown in a simplified flow chart in figure 10.

For more detailed information on the program organization and logic, please see the comment statements in the listing of the program source code in the Appendix.

PROGRAM APPLICATION

STEREO is written in FORTRAN IV language. It was originally developed for use on a PRIME computer, but should execute successfully using any standard FORTRAN compiler. A listing of the program is given in Appendix 1, and tables summarizing the function and parameters of some key subroutines are in Appendix 2.

Some changes may be necessary to adapt STEREO to different computer systems. The plotting subroutine (SUBROUTINE SPLOT) used in program STEREO, and listed in the Appendix, is designed to use a Zeta plotter and Zeta FORTRAN compatible subroutines. Users with other systems must supply their own plotting subroutine, named "SPLOT", which is appropriate for their own computer's plotting hardware and software. For more information on the plotting routine, see the comment statements in the listing of subroutine SPLOT.

Also, the input file unit number, variable 'IFILE' in the main program, may need to be set to a value other than 1.

INPUT PARAMETERS

To generate a stereogram, the user inputs the locations of a sequence of points in three dimensions, and for each point instructs the program either to draw to the point, to move to the point without drawing, or to draw a three dimensional symbol at the point. The program then constructs a figure consisting of line segments and point representations.

The point representations, called "jacks", are three dimensional symbols generated by the program, and they consist of three orthogonal line segments intersecting at a point. The three legs of each jack are oriented north-south, east-west, and radially, or, if the input is in cartesian coordinates, the legs are oriented parallel to the coordinate axes. The jacks can vary in size according to a magnitude parameter assigned to each point by the user.

The user must also inform the program the direction from which he wishes to view the figure. This is done by inputing the location of the view point, the point from which he is looking, and the location of the center point, the point which he is looking toward.

Several additional parameters are required. Table 1 lists the required and optional input parameters, and the page where each is explained in the following summary.

SUMMARY OF MEANING AND FORMAT OF INPUT VARIABLES

The locations of points input to the program may all be in spherical, or all in cartesian coordinates. The following summary, which explains the meaning and format of the input variables, is divided into two sections. The first section explains the entire program input when using spherical coordinates, and the second section explains the entire input when cartesian coordinates are used. In each section, the variables are listed in the order they are read by the program.

Table 1

		Page of Explanation	
	Parameter	Spherical	Cartesian
	1. VIEW POINT	20	23
	VIEW VECTOR	20	24
	2. CENTER POINT	20	24
	3. PLOT HEIGHT	21	24
	4. PLOT WIDTH	21	24
	5. EYE SEPARATION	21	24
	6. PLOT FOCAL LENGTH	21	25
*	7. COORDINATE SYSTEM		25
*	8. JACK SIZE DISTRIBUTION	22	25
*	9. SCALE	22	25
*	10. DEPTH OF FIELD	22	26
	11. DATA FLAG	22	26
	12. DATA LIST	22	26

* = OPTIONAL

INPUT FORMAT WHEN USING SPHERICAL COORDINATES

1. View - the point in space from which the observer is looking. The view parameters can be input either as the coordinates of the view point, or as a vector from the center point to the view point.

To input view as a point, use the following format:

column	name	format	explanation
12-20	VIEW(1)	F9.3	Latitude of view point, in decimal degrees
21-30	VIEW(2)	F10.3	Longitude of view point, in decimal degrees
31-40	VIEW(3)	F10.3	Elevation of view point, in decimal degrees

To input view as a vector from the center point, use the format below:

column	name	format	explanation
6-8	IVIEW	A2	Indicates this is a vector view when IVIEW = 'VE'
12-20	VIEW(1)	F9.3	Altitude--angle of view, in decimal degrees above the plane tangent to the surface of the sphere at the center point. (see figure 11a)
21-30	VIEW(2)	F10.3	Aximuth of view with respect to center point. (see figure 11a) if VIEW is from the north, azimuth = 0.0 if VIEW is from the east, azimuth = 90.0 if VIEW is from the southeast, azimuth = 135.0 etc.
31-40	VIEW(3)	F10.3	Distance between view point and center point, in kilometers.

2. Center point - the point in space which the observer is looking directly toward.

column	name	format	explanation
11-20	CENTER(1)	F10.3	Latitude of center point in decimal degrees
21-30	CENTER(2)	F10.3	Longitude of center point in decimal degrees
31-40	CENTER(3)	F10.3	Elevation of center point in kilometers

3. Plot height

column	name	format	explanation
11-20	PLTHT	F10.3	Height of plot in inches

4. Plot width - width of plot; points outside the bounds specified by plot height and width will not be plotted.

column	name	format	explanation
11-20	PLTWD	F10.3	Width of plot in inches; if PLTWD > EYESEP, left and right halves of plot may overlap

5. Eye separation - distance between the eyes (interocular spacing); average is about 2½ inches.

column	name	format	explanation
11-20	EYESEP	F10.3	Eye separation in inches

6. Plot focal length - apparent distance from the plot to the viewer's eye. When using optical devices to view stereo plots, the plot focal length can be calculated as follows:

$$\text{plot focal length} = \frac{\text{viewing device focal length}}{\text{viewing device magnification}}$$

for the standard pocket stereoscope, it is 2.5 inches.

column	name	format	explanation
11-20	PFOCAL	F10.3	Plot focal length in inches

7. Jack size distribution (optional) - distribution of sizes of "jack" symbols with respect to magnitude; up to 10 lines of separate distributions can be used.

column	name	format	explanation
6-10	RMAGLO	F5.2	Lower magnitude bound
11-15	RMAGUP	F5.2	Upper magnitude bound
16-20	SLO	F5.2	Size in inches at lower bound
21-25	SUP	F5.2	Size in inches at upper bound

RMAGLO = 999.0 to terminate size distribution input.

8. Scale (optional) - scale of plot at center point, expressed as the ratio of distance on the earth to unit distance on the plot.

column	name	format	explanation
11-20	SCALE	G10.3	Scale of plot at center point

9. Depth of field (optional) - the distance along the direction of view between the nearest and farthest points. It is used by the program to optimize the stereo base length.

column	name	format	explanation
11-20	RANGE	F10.3	Distance range

10. Data flag - indicates that all parameters have been input, and data points follow.

column	name	format	explanation
11-20	KOMAND	A2	When KOMAND = 'PL', data list follows

11. Data list - a list of points to be plotted in sequence.

column	name	format	explanation
1-10	COORD1	F10.3	Latitude of point in decimal degrees
11-20	COORD2	F10.3	Longitude of point in decimal degrees

21-30	COORD3	F10.3	Elevation of point in kilometers
31-40	RMAG	F10.3	Magnitude, for determining jack size
41-50	IPEN	I1	Type of point, as follows: = 1, draw a three dimensional jack symbol at this point = 2, draw a continuous line to this point = 3, move to this point without drawing

INPUT FORMAT WHEN USING CARTESIAN COORDINATES

1. View - the point in space from which the observer is looking. The view parameters can be input either as the coordinates of the view point, or as a vector from the view point to the center point. (For an explanation of the center point, see "Center point", below.)

To input view as a point use the following format:

column	name	format	explanation
12-20	VIEW(1)	F9.3	value of x-component of point location
21-30	VIEW(2)	F10.3	value of y-component of point location
31-40	VIEW(3)	F10.3	value of z-component of point location

To input view as a vector from the center point, using the following format:

column	name	format	explanation
6-8	IVIEW	A2	Flag indicating this is a vector view must be = 'VE'
12-20	VIEW(1)	F9.3	Altitude; angle of view above plane passing through center point parallel to x,z plane (see figure 11b)

21-30	VIEW(2)	F10.3	Azimuth of view with respect to center point if azimuth = 0.0, view is from the +x direction if azimuth = 90.0, view is from the +z direction etc. (see figure 11b)
31-40	VIEW(3)	F10.3	Distance between view point and center point

2. Center point - the point in space which the observer is looking directly toward.

column	name	format	explanation
11-20	CENTER(1)	F10.3	value of x-component of point location
21-30	CENTER(2)	F10.3	value of y-component of point location
31-40	CENTER(3)	F10.3	value of z-component of point location

3. Plot height

column	name	format	explanation
11-20	PLTHT	F10.3	Height of plot in inches

4. Plot width - width of plot. Points outside the bounds specified by plot height and width will not be plotted.

column	name	format	explanation
11-20	PLTWD	F10.3	Width of plot in inches; if PLTWD > EYESEP left and right halves of plot may overlap

5. Eye separation - distance between the eyes (interocular spacing); average is about 2.5 inches.

column	name	format	explanation
11-20	EYESEP	F10.3	Eye separation in inches

6. Plot focal length - apparent distance from the plot to the viewer's eyes. When using optical devices to view stereo plots, the plot focal length can be calculated as follows:

$$\text{plot focal length} = \frac{\text{viewing device focal length}}{\text{viewing device magnification}}$$

For the standard pocket stereoscope, it is 2.5 inches

column	name	format	explanation
11-20	PFOCAL	F10.3	Plot focal length in inches

7. Coordinate system - command indicating that input is in cartesian coordinates.

column	name	format	explanation
1-2	KOMAND	A2	KOMAND must be 'CA', for CArtesian

8. Jacksize distribution (optional) - distribution of sizes of "jack" symbols with respect to magnitude. Up to 10 independent distributions can be used. If no size is specified, all jacks will be 0.1 inches, by default.

column	name	format	explanation
6-10	RMAGLO	F5.2	Lower magnitude bound
11-15	RMAGUP	F5.2	Upper magnitude bound
16-20	SLO	F5.2	Size in inches at lower bound
21-25	SUP	F5.2	Size in inches at upper bound

RMAGLO = 999.0 etc.

9. Scale (optional) - scale of plot at center point, expressed as

column	name	format	explanation
11-20	SCALE	G10.3	Scale of plot at center point

10. <u>Depth of field (optional)</u> - the distance along the direction of view between the nearest and farthest point. It is needed only when plotting figures with depths of field much larger or much smaller than their width.

column	name	format	explanation
11-20	RANGE	F10.3	Distance range

11. <u>Data flag</u> - indicates that all parameters have been input, and data points follow.

column	name	format	explanation
11-20	KOMAND	A2	Flag indicating that data list follows, must be = 'PL'

12. <u>Data list</u> - a list of points to be plotted in sequence.

column	name	format	explanation
1-10	COORD1	F10.3	Value of x-component of point location
11-20	COORD2	F10.3	Value of y-component of point location
21-30	COORD3	F10.3	Value of z-component of point location
31-40	RMAG	F10.3	Magnitude of point, for determining jack size.
41-50	IPEN	I10	Type of point, as follows: = 1, draw a three dimensional jack symbol at this point = 2, draw a continuous line to this point = 3, move to this point without drawing

VIEWING METHOD

The method of viewing the stereograms is constrained by the parallel projection method used to generate them. To view the stereograms without distortion, both plots of a stereo pair must lie in the same plane. Also, the optical path from the center of each plot to the corresponding eye must be perpendicular to the plot surface, and the apparent distance from the plot surface to the observer's eyes should be equal to the plot focal length.

This means that the two input parameters controlling the viewing geometry--eye separation and plot focal length--must be chosen according to the type of stereoscope used to view the stereograms. Many different types of stereoscopes are available for viewing stereo photographs, and any of these are suitable for viewing the stereograms generated by STEREO as long as the above conditions are maintained. The most common viewing device, the pocket stereoscope, requires that the eye separation parameter be equal to the observer's inter-ocular spacing (average is about 2.5 inches), and the plot focal length should be equal to about 2.5 inches.

Slight deviation from ideal conditions will not noticeably distort the stereo image, so meticulous adherance to viewing conditions is not essential. Results will be good as long as viewing parameters are within about 10% of ideal. For example, when a stereogram will be viewed by several people with different interocular spacings, the stereogram can be plotted at an average interocular value without significant distortion for any of the viewers.

DISCUSSION

The stereograms generated by the program, using the parallel projection method described, are an exact representation of the geometrical perspective of the figure being projected, because each point is projected along the line of sight from the projection point. The combination of the different perspectives from the two carefully selected projection points, when properly viewed, recreates an undistorted image of the original figure.

Stereograms do not look exactly the same as the real figure, however, because there is important information perceived when observing real objects which cannot be reproduced on a two-dimensional plot. When viewing real objects, the eyes must change focus when looking at points different distances away, and this fact is used as a cue to distances in three-dimensional perception. All the points in a stereogram are actually the same distance from the eyes, so the eye's focus does not change. The fact that the angles of convergence change when viewing points in a stereogram while the focus remains the same can be confusing to some peoples innate three-dimensional perception, and may be a source of eye strain (Rule, 1941).

Another cue in distance perception is the known size of familiar objects, which is used to help determine the absolute distance to an object (Boutry, 1965, p. 309). Stereograms used for scientific illustration will commonly contain no objects of known size, so the absolute distance to the figure may be difficult to determine, and the stereo image may appear to float in front of the observer at an indeterminate distance.

However, for the perception of relative distances between objects, geometrical and binocular perspectives are by far the most important cues (Fry, 1965, p. 74), and these cues are rarely misinterpreted (Yellot, 1981). STEREO generates stereograms with accurately represented perspectives, so that relative distances, slopes, angles, or other spatial relationships seen in the stereo image will be perceived correctly by all people with normal stereoptic vision.

Acknowledgment

We would like to thank Douglas Given and Paul Reasenberg for their helpful comments and suggestions.

REFERENCES

Boutry, G.A., 1962, Instrumental Optics [trans. R. Auerbach]: New York, Interscience Publishers, Inc., 535 p.

Dudley, L.P., 1965, Stereoscopy, in Kingslake, R.,ed, Applied Optics and and Optical Engineering, vol 2: New York, Academic Press, p. 77-117.

Fry, G.A., 1965, The Eye and Vision, in Kinglake, R., ed, Applied Optics and Optical Engineering, vol. 2: New York, Academic Press, p. 1-76.

Johnson, C.E., 1979, Seismotectonics of the Imperial Valley of southern California, pt. 2 of thesis: California Institute of Technology, Pasadena, California, p. 122-323.

Johnson, C.E., and Richter, F.M., 1979, Stereoviews of Seismicity Associated with Subduction Zones: the Journal of Geology, vol 87, no. 5, p. 467-474.

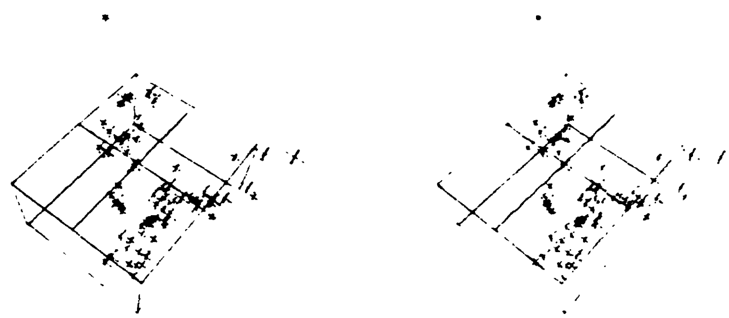
Rule, J.T., 1941, The Geometry of Stereoscopic Projection: Journal of the Optical Society of America, vol. 31, p. 325-334.

Yellott, J.I., Jr., 1981, Binocular Depth Inversion: Scientific American, vol. 245, no. 1, p. 148-159.

APPENDICES

1. Program source code listing
2. Explanation of key subroutines
3. Example of input and output

EXAMPLE 1. Aftershocks of the 1971 San Fernando earthquake, between magnitudes 3.0 and 4.0.



INPUT:

```
VIEW              34.6     -118.6       40.0
CENTER            34.4     -118.4       -8.0
PLOTHEIGHT         3.0
PLOTWIDTH          2.4
EYE SEP            2.4
FOCAL             2.25
SIZE
     3.0  3.5  0.05 0.05
     3.5  4.0  0.10 0.10
     999.
PLOT
    34.309  -118.479   -18.600      3.400          1
    34.341  -118.483    -8.700      3.800          1
    34.346  -118.289   -12.500      3.800          1
    34.354  -118.469    -9.700      3.500          1
    34.314  -118.489   -11.500      3.800          1
    34.352  -118.441   -10.700      3.800          1
    34.415  -118.447   -11.900      3.400          1
    34.432  -118.421   -12.300      3.200          1
    34.332  -118.312    -8.600      3.400          1
    34.428  -118.401    -6.900      3.200          1
    34.441  -118.458   -11.100      3.300          1
    34.373  -118.366    -6.500      3.500          1
    34.353  -118.325    -6.900      3.500          1
    34.417  -118.439    -8.000      3.200          1
    34.293  -118.522   -10.900      3.400          1
    34.445  -118.447   -10.300      3.200          1
    34.457  -118.442   -12.400      3.300          1
    34.397  -118.363    -8.500      3.300          1
    34.337  -118.314    -0.500      3.500          1
    34.296  -118.321    -2.400      3.500          1
    34.374  -118.438   -12.400      3.600          1
    34.392  -118.445   -10.900      3.700          1
    34.448  -118.429   -11.000      3.300          1
    34.352  -118.337    -4.600      3.700          1
    34.357  -118.339    -7.600      3.200          1
    34.412  -118.438    -7.600      3.300          1
    34.359  -118.457    -7.600      3.400          1
    34.407  -118.431    -9.700      3.400          1
    34.411  -118.429    -9.600      3.900          1
    34.319  -118.306    -2.300      3.300          1
    34.374  -118.376     2.000      3.100          1
    34.250  -118.190     0.700      3.100          1
    34.299  -118.511    -1.200      3.800          1
    34.321  -118.290    -4.300      3.400          1
    34.358  -118.466    -7.800      3.200          1
    34.471  -118.411   -12.900      3.900          1
    34.407  -118.441    -7.900      3.400          1
    34.427  -118.435    -9.700      3.400          1
    34.287  -118.544    -0.600      3.500          1
    34.402  -118.453    -9.600      3.300          1
    34.334  -118.298   -10.000      3.100          1
    34.346  -118.492     2.000      3.000          1
    34.358  -118.306    -5.300      3.500          1
    34.383  -118.448    -6.200      3.000          1
```

Example 1 input, continued.

34.347	-118.347	-5.200	3.000	1
34.398	-118.350	-12.100	3.200	1
34.386	-118.429	-4.400	3.200	1
34.374	-118.427	-7.300	3.200	1
34.454	-118.406	-15.200	3.600	1
34.398	-118.426	-4.900	3.500	1
34.398	-118.445	-6.200	3.500	1
34.310	-118.316	-6.400	3.000	1
34.396	-118.328	-2.900	3.500	1
34.441	-118.411	-10.800	3.700	1
34.408	-118.368	-11.100	3.100	1
34.429	-118.453	-12.700	3.200	1
34.336	-118.363	2.000	3.500	1
34.419	-118.383	-7.500	3.500	1
34.451	-118.450	-12.900	3.300	1
34.449	-118.438	-12.000	3.000	1
34.350	-118.286	-5.500	3.000	1
34.376	-118.433	-4.900	3.900	1
34.377	-118.437	-0.800	3.300	1
34.446	-118.448	-2.800	3.000	1
34.468	-118.461	-11.300	3.300	1
34.355	-118.473	-5.700	3.700	1
34.395	-118.429	-7.300	3.500	1
34.284	-118.528	-3.000	4.000	1
34.433	-118.403	-4.800	3.600	1
34.379	-118.431	-7.800	3.000	1
34.277	-118.505	-12.200	3.500	1
34.366	-118.492	-5.100	3.000	1
34.366	-118.492	-5.100	3.000	1
34.3	-118.4	0.0		3
34.5	-118.4	0.0		2
34.5	-118.3	0.0		2
34.3	-118.3	0.0		2
34.3	-118.5	0.0		2
34.5	-118.5	0.0		2
34.5	-118.4	0.0		2
34.4	-118.5	0.0		3
34.4	-118.3	0.0		2
34.3	-118.3	0.0		3
34.3	-118.3	-10.0		2
34.5	-118.3	-10.0		2
34.5	-118.5	-10.0		2
34.3	-118.5	-10.0		2
34.3	-118.3	-10.0		2
34.5	-118.3	0.0		3
34.5	-118.3	-10.0		2
34.5	-118.5	0.0		3
34.5	-118.5	-10.0		2
34.3	-118.5	0.0		3
34.3	-118.5	-10.0		2

JACK (P1,P2,P3, SIZE)

JACK is used to draw a three dimensional "jack" symbol (3 orthogonal line segments intersecting at their midpoints) at the specified location.

Argument	Type	Value	Explanation
P1,P2,P3	REAL		Coordinates of point at which a three dimensional "jack symbol is drawn. Coordinates may be either spherical or cartesian, as set by ISPHER in call to SETUP if ISPHER = 0 P1,P2,P3 = x,y,z coordinates of point if ISPHER = 1 P1 = latitude in decimal degrees P2 = longitude in decimal degrees P3 = elevation in kilometers
SIZE	REAL		Size in inches of "jack in plane of center point. Jacks closer to view point will be larger; jacks farther away will be smaller.

DRAW3D (P1, P2, P3)

DRAW3D gives the user direct control over pen movement. It is used to draw a continuous line from one point to another in three dimensions.

Argument	Type	Value	Explanation
P1,P2,P3	REAL		Coordinates of point to which the pen draws from previous position. Coordinates may be either spherical or cartesian, as set by ISPHER in initial call to SETUP if ISPHER = 1 (spherical coordinates) P1 = latitude in decimal degrees P2 = longitude in decimal degrees P3 = elevation in kilometers if ISPHER = 0 P1,P2,P3 = cartesian coordinates of point

MOVE3D

MOVE3D gives the user direct control over pen movement. It is used to move from one point to another in three dimensions with the pen up (i.e. without drawing)

Argument	Type	Value	Explanation
P1,P2,P3	REAL		Coordinates of point to which pen moves without drawing Coordinates are either spherical or cartisian, as set by ISPHER in initial call to SETUP if ISPHER = 1 (spherical coordinates) P1 = latitude in decimal degrees P2 = longitude in decimal degrees P3 = elevation in kilometers if ISPHER = 0 P1,P2,P3 = cartesian coordinates of point

SETUP (VIEW, CENTER, PLTH, PLTW, PLTS, SANGLE, FOCAL, ISPHER)

SETUP must be the first stereo subroutine called, and should be called only once. It initializes COMMON variables, and sets up vectors used in projection subroutine PROJXN.

Argument	Type	Value	Explanation
VIEW	REAL ARRAY		Coordinates of viewpoint, either cartesion or spherical as set by ISPHER if ISPHER = 0 VIEW(1), VIEW(2), VIEW(3) = Cartesian coordinates of center point if ISPHER = 1 VIEW(1) = latitude in decimal degrees VIEW(2) = longitude in decimal degrees VIEW(3) = elevation in kilometers
CENTER	REAL ARRAY		Coordinates of center point, either cartesian or spherical, as set by ISPHER if ISPHER = 0 CENTER(1), CENTER(2), CENTER(3) = cartesian coordinates of center point if ISPHER = 1 CENTER(1) = latitude in decimal degrees CENTER(2) = longitude in decimal degrees CENTER(3) = elevation in kilometers
PLTH	REAL		Height of plot in inches
PLTW	REAL		Width of plot in inches
PLTS	REAL		Distance in inches between centers of left and right plots
SANGLE	REAL		Angle, in degrees, subtended at the center point by the viewpoint and projection point
FOCAL	REAL		Focal length in inches of device used to view plots
ISPHER	INTEGER	0	VIEW, CENTER, and points to be plotted are in cartesion coordinates
		1	VIEW, CENTER, and points to be plotted are in spherical coordinates

APPENDIX 1

Program Source Code Listing

```
C  STEREO PLOTTING DRIVER AND SUBROUTINES
C
C        BLAME: PETER GERMAN
C               USGS, OFFICE OF EARTHQUAKE STUDIES
C               DECEMBER 19, 1980
C
C  FOR EXPLANATION OF SOME KEY VARIABLES, SEE COMMENTS AT
C  BEGINNING OF SUBROUTINE 'SETUP'
C
      DIMENSION VIEW(3),CENTER(3),RMAGUP(10),RMAGLO(10),SUP(10),SLO(10)
      DATA KPLOT/'PL'/,KSIZE/'SI'/,KAND/'  '/,KRANG/'RA'/,KARTES/'CA'/
      DATA KSCALE/'SC'/,SCALE/-1.0/
      CALL OPENOT(7,'SPLOT!')
      CALL PLOTS (0.0,0.0,7)
      CALL SPEED (10)
C-----IFILE = INPUT FILE UNIT NUMBER
      IFILE = 1
C-----DEFAULT JACK SIZE (0.1 INCH)
      ISPHER = 1
      N = 0
      RANGE = -1.0
C-----READ PLOTTING PARAMATERS
      READ (IFILE,375) IVIEW, VIEW
  375 FORMAT (5X,A2,4X,F9.3,2F10.3)
      READ (IFILE,100) CENTER
  100 FORMAT (10X,3F10.3)
      READ (IFILE,200) PLTHT
      READ (IFILE,200) PLTWD
      READ (IFILE,200) EYESEP
      READ (IFILE,200) PFOCAL
  200 FORMAT (10X,F10.3)
C-----FOLLOWING LOOP READS AND INTERPRETS OPTIONAL INPUT PARAMETERS
    2 READ (IFILE,350) KOMAND,VARIBL
  350 FORMAT (A2,8XG10.3)
      IF (KOMAND.EQ.KPLOT) GO TO 7
      IF (KOMAND.EQ.KSIZE) GO TO 4
      IF (KOMAND.EQ.KRANG) GO TO 5
      IF (KOMAND.EQ.KSCALE)GO TO 6
      IF (KOMAND.EQ.KARTES) ISPHER = 0
      GO TO 2
C-----FOLLOWING LOOP READS IN MAGNITUDE-SYMBOL SIZE DISTRIBUTION
    4 READ (IFILE,400) RMAGLO(N),RMAGUP(N),SLO(N),SUP(N)
  400 FORMAT(4F5.2)
      IF (RMAGLO(N).GT.998.0) GOTO 2
      N = N + 1
      GO TO 4
    5 RANGE = VARIBL
      GO TO 2
    6 FACTR = 1.0
      IF(ISPHER.EQ.1) FACTR = 2.54E-5
```

```
      SCALE = FACTR*VARIBL
      GO TO 2
    7 CALL SETUP(VIEW,IVIEW,CENTER,PLTHT,PLTWD,EYESEP,RANGE,
     *           PFOCAL,ISPHER,SCALE)
C-----READ AND PLOT DATA LIST
   10 READ(IFILE,500,END=20,ERR=20) COORD1,COORD2,COORD3,RMAG,IPEN
  500 FORMAT (4F10.3,I10)
      IF (IPEN.EQ.3) CALL MOVE3D(COORD1,COORD2,COORD3)
      IF (IPEN.EQ.2) CALL DRAW3D(COORD1,COORD2,COORD3)
      IF (IPEN.NE.1) GO TO 10
      IF (N.EQ.0) GO TO 50
C-----CALCULATE SIZE OF JACK
      DO 30 I = 1,N
      K = I
      IF (RMAG.LT.RMAGUP(I).AND.RMAG.GE.RMAGLO(I)) GO TO 40
   30 CONTINUE
      GO TO 10
   40 SIZE=(RMAG-RMAGLO(K))*(SUP(K)-SLO(K))/(RMAGUP(K)-RMAGLO(K))+SLO(K)
C-----PLOT JACK
   50 CALL JACK(COORD1,COORD2,COORD3,SIZE)
      GO TO 10
   20 CALL FINISH
      STOP
      END
```

```
      SUBROUTINE SETUP(VIEW,IVIEW,CENTER,PLTH,PLTW,EYES,
                       RANGE,PFOCAL,ISPHER,SCALE)
C
C       BY PETER GERMAN
C          USGS, OFFICE OF EARTHQUAKE STUDIES
C          DECEMBER 19, 1980
C
C THIS SUBROUTINE INITIALIZES VARIABLES, AND SETS UP
C VECTORS USED BY THE PROJECTION SUBROUTINE 'PROJXN'
C
C  VIEW   - THE VIEW POINT OR VECTOR (A THREE ELEMENT ARRAY)
C           THE POINT OR DIRECTION FROM WHICH THE VIEWER IS
C           LOOKING.
C  IVIEW  - INDICATES TYPE OF VIEW, POINT OR VECTOR
C           = 'VE' FOR VECTOR (ALTITUDE, AZIMUTH, DISTANCE)
C           = ANYTHING FOR POINT VIEW
C  CENTER - THE CENTER POINT (A THREE ELEMENT ARRAY)
C  PLTH   - PLOT HEIGHT IN INCHES
C  PLTW   - PLOT WIDTH IN INCHES
C  EYESEP - EYE SEPARATION (INTER-OCULAR SPACING) IN INCHES
C  RANGE  - DISTANCE RANGE; THE DISTANCE ALONG THE VIEW DIRECTION
C           BETWEEN THE NEAREST AND FARTHEST POINTS BEING PLOTTED
C  PFOCAL - PLOT FOCAL LENGTH, WHICH IS THE APPARENT DISTANCE FROM
C           THE PLOT TO THE VIEWER'S EYES
C  ISPHER - FLAG INDICATING TYPE OF COORDINATE SYSTEM
C         = 0, FOR CARTESIAN COORDINATES
C         = 1, FOR SPHERICAL COORDINATES
C  CX,CY,CZ - CARTESIAN COORDINATES OF THE CENTER POINT
C  CLSX,CLSY,CLSZ - CARTESIAN COORDINATES OF THE CENTRAL LINE
C           OF SIGHT VECTOR, WHICH IS THE VECTOR FROM THE
C           VIEW POINT TO THE CENTER POINT
C  CO     - VECTOR FROM CENTER POINT TO ORIGIN (A THREE
C           ELEMENT ARRAY)
C  ALPHA  - THE ANGLE BETWEEN THE X,Y PROJECTION OF THE
C           LINE OF SIGHT VECTOR AND THE -Y AXIS
C  BETA   - THE ANGLE BETWEEN THE LINE OF SIGHT VECTOR AND
C           THE -Z AXIS
C  GAMMA  - THE ANGLE BETWEEN THE X,Y PROJECTION OF THE
C           CENTER TO ORIGIN VECTOR AND THE -Y AXIS
C  H      - DISTANCE BETWEEN VIEW POINT AND CENTER POINT
C  BASELN - ONE HALF THE STEREO BASE LENGTH, WHICH IS THE DISTANCE
C           BETWEEN THE TWO PROJECTION POINTS
C  SCALE  - SCALE OF PLOT AT CENTER POINT
C
      COMMON/VECTOR/CX,CY,CZ,ALPHA,BETA,GAMA,H,BASELN
      COMMON/PLT/PLTHT,PLTWD,EYESEP,SCAL
      COMMON/COUNT/ N,KP
      COMMON/COORD/ ISPHR
      DIMENSION VIEW(3),CENTER(3),CO(3)
      DATA PI/3.1415927/,N/0/,IVECT/'VE'/
```

```
      PLTHT = PLTH
      PLTWD = PLTW
      EYESEP = EYES
      GAMA = 0.0
      ISPHR = ISPHER
      IF (ISPHER.EQ.1) GO TO 5
C-----FIND CLS FOR CARTESIAN COORDINATES
          CX = CENTER(1)
          CY = CENTER(2)
          CZ = CENTER(3)
          IF (IVIEW.EQ.IVECT) GO TO 7
          CLSX = CX - VIEW(1)
          CLSY = CY - VIEW(2)
          CLSZ = CZ - VIEW(3)
          GO TO 10
C-----FIND CLS FOR SPHERICAL COORDINATES
C-----RADIUS OF EARTH = 6371.3 KM
    5 CLAT = D2R(CENTER(1))
      CLON = D2R(CENTER(2))
      CR = CENTER(3) + 6371.3
      CALL POLCAT(CLAT,CLON,CR,CX,CY,CZ)
      IF (IVIEW.EQ.IVECT) GO TO 7
      VLAT   = D2R(VIEW(1))
      VLON   = D2R(VIEW(2))
      VR     = VIEW(3) + 6371.3
      CALL POLCAT(VLAT,VLON,VR,VX,VY,VZ)
      CLSX = CX - VX
      CLSY = CY - VY
      CLSZ = CZ - VZ
      GO TO 10
C-----FIND CLS FOR VECTOR VIEW
    7 ALT = D2R(VIEW(1))
      AZM = D2R(VIEW(2))
      H = VIEW(3)
      IF (SCALE.GT. 0.0) H = SCALE*PFOCAL
      IF (SCALE.LT. 0.0) SCALE = H/PFOCAL
      A = COS(ALT) * COS(AZM)
      B = SIN(ALT)
      C = COS(ALT) * SIN(AZM)
      IF(ISPHER.EQ.1.AND.CR.GT.1.0.AND.ABS(CLAT).LT.89.9) GO TO 8
      CLSX = -A*H
      CLSY = -B*H
      CLSZ = -C*H
      GO TO 10
    8 CLA = COS(CLAT)
      CLO = COS(CLON)
      SLA = SIN(CLAT)
      SLO = SIN(CLON)
      CLSX = -1.0*(B*CLA*CLO - A*SLA*CLO - C*SLO)*H
      CLSY = -1.0*(A*CLA + B*SLA)*H
```

```
      CLSZ = -1.0*(A*SLA*SLO - B*CLA*SLO - C*CLO)*H
C-----TEST CLSX,CLSY BEFORE USING ATAN FUNCTION
   10 IF (ABS(CLSY).GT.0.01.OR.ABS(CLSX).GT.0.01) GO TO 15
      ALPHA = 0.0
      BETA  = 0.0
      GO TO 17
   15 ALPHA = ATAN2(CLSY,CLSX) - PI/2.0
      BETA  = ATAN2(SQRT(CLSX**2 + CLSY**2) , -1.0*CLSZ)
C-----SETUP CENTER TO ORIGIN VECTOR
      CO(1) = -1.0*CX
      CO(2) = -1.0*CY
      CO(3) = -1.0*CZ
C-----FIND GAMMA FROM CENTER TO ORIGIN VECTOR
      CALL ROTATE(1,CO,ALPHA,BETA,0.0)
      IF (ABS(CO(1)).GT.0.1.OR.ABS(CO(2)).GT.0.1) GO TO 16
C-----IF VIEW POINT, CENTER POINT, AND ORIGIN ARE CO-LINEAR
C-----SETUP ARTIFICIAL ORIGIN TO FIND GAMMA AND ORIENT THE Y AXIS
      CO(1) =0.0
      CO(2) = -100.0
      CO(3) = 0.0
      CALL ROTATE(1,CO,ALPHA,BETA,0.0)
   16 GAMA = GAMA + ATAN2(CO(2),CO(1)) + PI/2.0
C-----NOW COMPUTE H, BASELN, SCALE
      IF(IVIEW.EQ.IVECT) GO TO 18
      IF(SCALE.LT.0.0) GO TO 17
      H = SCALE*PFOCAL
      GO TO 18
   17 H = SQRT(CLSX**2 + CLSY**2 + CLSZ**2)
      SCALE = H/PFOCAL
   18 IF (RANGE.GT.0.0) GO TO 20
      BASELN = H * 0.034920769
C-----0.034290769 IS TAN( 2.0 DEGREES )
      GO TO 25
C-----OPTIMIZE STEREO BASE, DEPENDING ON RANGE OF DEPTH
   20 RBY2 = RANGE/2.0
      BASELN = ((H-RBY2)*(H+RBY2))/(100.0*RANGE)
      IF(ATAN(BASELN/(H-RBY2)).GT.D2R(7.0)) BASELN = H * TAN(D2R(7.0))
   25 SCAL = SCALE
      RETURN
      END
```

```
      SUBROUTINE MOVE3D(P1,P2,P3)
C
C     BY PETER GERMAN
C        USGS, OFFICE OF EARTHQUAKE STUDIES
C        DECEMBER 19, 1980
C
C  THIS SUBROUTINE GIVES USER DIRECT CONTROL OVER PLOTTER PEN
C  BY CAUSING PEN TO MOVE TO COORDINATES OF POINT SPECIFIED
C  IN ARGUMENTS, WITH PEN UP (I.E. WITHOUT DRAWING)
C
      COMMON/COORD/ ISPHER
      IF (ISPHER.EQ.0) GO TO 10
      RLAT = D2R(P1)
      RLON = D2R(P2)
      R    = P3 + 6371.3
      CALL POLCAT(RLAT,RLON,R,X,Y,Z)
      CALL PAKRAT(X,Y,Z,0)
      RETURN
   10 CALL PAKRAT(P1,P2,P3,0)
      RETURN
      END
```

```
      SUBROUTINE DRAW3D(P1,P2,P3)
C
C     BY PETER GERMAN
C        USGS, OFFICE OF EARTHQUAKE STUDIES
C        DECEMBER 19, 1980
C
C  THIS SUBROUTINE GIVES USER DIRECT CONTROL OVER PLOTTER PEN
C  BY CAUSING PEN TO DRAW A CONTINUOUS LINE TO POINT SPECIFIED
C  IN ARGUMENTS.
C
      COMMON/COORD/ISPHER
      IF (ISPHER.EQ.0) GO TO 10
      RLAT = D2R(P1)
      RLON = D2R(P2)
      R    = P3 + 6371.3
      CALL POLCAT(RLAT,RLON,R,X,Y,Z)
      CALL PAKRAT(X,Y,Z,1)
      RETURN
   10 CALL PAKRAT(P1,P2,P3,1)
      RETURN
      END
```

```
      SUBROUTINE JACK(P1,P2,P3,SIZE)
C
C     BY PETER GERMAN
C        USGS, OFFICE OF EARTHQUAKE STUDIES
C        DECEMBER 19, 1980
C
C  THIS SUBROUTINE CREATES A 'JACK' SYMBOL CENTERED AT A POINT
C  SPECIFIED BY THE FIRST THREE CALLING ARGUMENTS (P1,P2,P3)
C
      COMMON/PLT/PLTHT,PLTWD,EYESEP,SCALE
      COMMON/COORD/ ISPHER
      DIMENSION P(3),DELTA(3),AL(3)
      P(1) = P1
      P(2) = P2
      P(3) = P3
      ARMSIZ = SIZE * SCALE/2.0
      IF (ISPHER.EQ.0) GO TO 20
      R    = P(3) + 6371.3
      P(1) = D2R( P(1) )
      P(2) = D2R( P(2) )
      IF (R.LT.1.0.OR.ABS(P1).GT.89.99) GO TO 15
      DELTA(1) = ATAN2(ARMSIZ,R)
      DELTA(2) = ATAN2(ARMSIZ,(R*COS(P(1))))
      DELTA(3) = ARMSIZ
      P(3)     = R/COS( DELTA(1) )
      DO 10 I=1,3
      IF (I.EQ.3) P(3) = R
      P(I) = P(I) + DELTA(I)
      CALL POLCAT(P(1),P(2),P(3),U,V,W)
      CALL PAKRAT(U,V,W,0)
      P(I) = P(I) - 2.0*DELTA(I)
      CALL POLCAT(P(1),P(2),P(3),U,V,W)
      CALL PAKRAT(U,V,W,1)
      P(I) = P(I) + DELTA(I)
   10 CONTINUE
      RETURN
   15 RLAT = P(1)
      RLON = P(2)
      CALL POLCAT(RLAT,RLON,R,P(1),P(2),P(3))
   20   DO 30 I = 1,3
        P(I) = P(I) + ARMSIZ
        CALL PAKRAT(P(1),P(2),P(3),0)
        P(I) = P(I) - 2.0*ARMSIZ
        CALL PAKRAT(P(1),P(2),P(3),1)
        P(I) = P(I) + ARMSIZ
   30   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PAKRAT(X,Y,Z,LPEN)
C
C     BY PETER GERMAN
C        USGS, OFFICE OF EARTHQUAKE STUDIES
C        DECEMBER 19, 1980
C
C  THIS SUBROUTINE PACKS THREE DIMENSIONAL POINTS
C  INTO A ONE DIMENSIONAL BUFFER, THEN SENDS THEM TO
C  THE PROJECTION SUBROUTINE 'PROJXN' WHEN THE BUFFER
C  IS FULL
C
      COMMON/COUNT/ N,KP
      COMMON/POINT/ P(3000),IPEN(1000)
      MAX = 1000
      N = N + 1
      P(3*N-2) = X
      P(3*N-1) = Y
      P(3*N  ) = Z
      IPEN(N)  = LPEN
C-----TEST BUFFER - IF FULL, SEND IT TO PROJXN
      IF (N.EQ.MAX) GO TO 10
      RETURN
   10 CALL PROJXN
C-----SETUP FIRST POINT IN NEXT BUFFER
      P(1) = X
      P(2) = Y
      P(3) = Z
      IPEN(1) = 0
      N = 1
      RETURN
      END
```

```
      SUBROUTINE ROTATE(N,P,ALPHA,BETA,GAMMA)
C
C     BY PETER GERMAN
C        USGS, OFFICE OF EARTHQUAKE STUDIES
C        DECEMBER 19, 1980
C
C  THIS SUBROUTINE ROTATES AN ARRAY OF 'N' POINTS P(X,Y,Z) BY
C  ANGLE ALPHA AROUND THE Z AXIS, THEN ANGLE BETA AROUND THE X
C  AXIS, AND FINALLY BY ANGLE GAMMA AROUND THE Z AXIS, IN A
C  RIGHT HANDED CARTESIAN COORDINATE SYSTEM.
C
      DIMENSION P(1)
      CA = COS(ALPHA)
      SA = SIN(ALPHA)
      CB = COS(BETA)
      SB = SIN(BETA)
      CG = COS(GAMMA)
      SG = SIN(GAMMA)
      R11 = CA * CG - SA * CB * SG
      R21 = -1.0*CA * SG - SA * CB * CG
      R31 = SA * SB
      R12 = SA * CG + CA * CB * SG
      R22 = CA * CB * CG - SA * SG
      R32 = -1.0*CA * SB
      R13 = SB * SG
      R23 = SB * CG
      R33 = CB
      DO 10 I = 1, N
        OLDX = P(3*I-2)
        OLDY = P(3*I-1)
        OLDZ = P(3*I)
        P(3*I-2) = OLDX*R11 + OLDY*R12 + OLDZ*R13
        P(3*I-1) = OLDX*R21 + OLDY*R22 + OLDZ*R23
   10   P(3*I) = OLDX*R31 + OLDY*R32 + OLDZ*R33
      RETURN
      END
```

```
      SUBROUTINE POLCAT(RLAT,RLON,R,X,Y,Z)
C
C THIS SUBROUTINE CONVERTS SPHERICAL COORDINATES TO CARTESIAN
C SUCH THAT (RLAT,RLON,R) BECOMES (X,Y,Z), WITH THE ORIGIN
C UNCHANGED, AND WITH THE +X AXIS PASSING THROUGH THE POINT
C LAT.,LON. = 0,0 ; THE +Y AXIS PASSING THROUGH 90,0 ; AND
C THE +Z AXIS PASSING THROUGH 0,-90
C
C RLAT = LATITUDE IN RADIANS
C RLON = LONGITUDE IN RADIANS
C R    = RADIUS
C
        X = R * COS(RLAT) * COS(RLON)
        Y = R * SIN(RLAT)
        Z = -1.0*R * COS(RLAT) * SIN(RLON)
      RETURN
      END
```

```
      SUBROUTINE TRANS(N,P,CX,CY,CZ)
C
C  THIS SUBROUTINE TRANSLATES ALL 'N' POINTS IN A CARTESIAN
C  COORDINATE SYSTEM, SUCH THAT THE ORIGIN TRANSLATES TO
C  THE POINT (CX,CY,CZ).
C
      DIMENSION P(1)
      DO 10 I = 1, N
         P(3*I-2) = P(3*I-2) - CX
         P(3*I-1) = P(3*I-1) - CY
   10    P(3*I)   = P(3*I)   - CZ
      RETURN
      END
```

```
      SUBROUTINE PROJXN
C
C     BY PETER GERMAN
C        USGS, OFFICE OF EARTHQUAKE STUDIES
C        PASADENA      DECEMBER 19, 1980
C
C  THIS SUBROUTINE PROJECTS POINTS IN THREE DIMENSIONS
C  ONTO A PLANE IN TWO DIMENSIONS.  IT USES A LINEAR
C  PROJECTION, DESIGNED TO BE EQUIVILENT TO A PHOTOGRAPH.
C
C KP    IS A COUNT OF THE NUMBER OF POINTS ACTUALLY PLOTTED
C ISIDE IS A FLAG INDICATING WHICH VIEW IS BEING PLOTTED
C       = 1, WHEN THE LEFT VIEW IS BEING PROJECTED AND PLOTTED
C       = 2, WHEN THE RIGHT VIEW IS BEING PROJECTED AND PLOTTED
C SHIFT = 0.0, FOR ALL POINTS ON THE LEFT SIDE OF PLOT
C       = EYESEP, FOR ALL POINTS ON THE RIGHT SIDE OF PLOT
C XBIG  IS THE MAXIMUM X VALUE THAT WILL FIT ON THE PLOT
C XSMAL IS THE MINIMUM X VALUE THAT WILL FIT ON THE PLOT
C
      COMMON/POINT/P(3000),IPEN(1000)
      COMMON/VECTOR/CX,CY,CZ,ALPHA,BETA,GAMMA,H,BASELN
      COMMON/COUNT/N,KP
      COMMON/PLT/PLTHT,PLTWD,EYESEP,SCALE
      INTEGER X,Y,Z
      KP = 0
      ISIDE = 0
      SHIFT = -0.5*EYESEP
      XBIG = PLTWD+SHIFT
      XSMAL= 0.0+SHIFT
C-----TRANSLATE ORIGIN OF COORDINATE SYSTEM TO THE CENTER
C-----POINT, THEN ROTATE IT SO THE Z-AXIS LIES ALONG THE
C-----LINE OF SIGHT.
      CALL TRANS(N,P,CX,CY,CZ)
      CALL ROTATE(N,P,ALPHA,BETA,GAMMA)
C-----PROJECT AND PLOT, LEFT SIDE FIRST
    5 BASELN = -1.0*BASELN
C-----SETUP FLAGS SO POINTS WILL BE PLOTTED ON PROPER SIDE OF PLOT
      ISIDE = ISIDE + 1
      KPOLD = -1
C-----THE FOLLOWING LOOP PLOTS ALL 'N' POINTS IN THE BUFFER, ON ONE
C-----SIDE OF THE PLOT (EITHER LEFT OR RIGHT)
          DO 10 I=1,N
          X = 3*I - 2
          Y = 3*I - 1
          Z = 3*I
C-----IF LAST POINT WAS NOT PLOTTED, LIFT PEN
          IF (KPOLD.EQ.KP) IPEN(I) = 0
          KPOLD = KP
C-----IF POINT IS 'BEHIND' VIEW POINT, DO NOT ATTEMPT TO PROJECT
          IF (P(Z).GE.H) GO TO 10
```

```
C-----THE FOLLOWING IS THE MEAT OF THE PROGRAM, WHEREIN A POINT
C-----IN THREE DIMENSIONS IS PROJECTED ONTO THE X,Y PLANE, USING
C-----A SIMPLE PERSPECTIVE PROJECTION FROM THE POINT (BASELN,0.0,H)
          PX = (P(X) - BASELN)*H/(H - P(Z))
          PY =  P(Y) * H/(H-P(Z))
C-----APPLY SCALE FACTOR, AND SHIFT POINT TO PROPER SIDE OF PLOT
          PX = PX/SCALE + PLTWD/2 + SHIFT
          PY = PY/SCALE + PLTHT/2
C-----IF POINT DOES NOT LIE WITHIN BOUNDARIES OF PLOT, DON'T PLOT
          IF (PX.GT.XBIG.OR.PX.LT.XSMAL) GO TO 10
          IF (PY.GT.PLTHT.OR.PY.LT.0.0) GO TO 10
C-----POINT HAS PASSED ALL TESTS, IS PLOTTED AND COUNTED
          CALL SPLOT(PX,PY,IPEN(I))
          KP = KP + 1
   10     CONTINUE
      IF (ISIDE.EQ.2) GO TO 15
C-----RESET VARIABLES FOR RIGHT SIDE OF PLOT
      SHIFT = 0.5*EYESEP
      XBIG  = PLTWD + SHIFT
      XSMAL = SHIFT
      GO TO 5
   15 N = 0
      RETURN
      END
```

```
      SUBROUTINE SPLOT (X,Y,IPEN)
C
C  THIS IS A DUMMY PLOT SUBROUTINE WHICH IS CALLED TO PLOT
C  EACH POINT.
C
C  X,Y  ARE THE COORDINATES OF EACH POINT TO BE PLOTTED, IN
C       INCHES FROM THE LOWER LEFT CORNER OF THE PLOT.
C  IPEN INDICATES WHETHER THE POINT IS TO BE DRAWN TO, OR MOVED TO
C       WITHOUT DRAWING.
C       WHEN IPEN = 0, THE POINT IS MOVED TO FROM THE PREVIOUS POINT
C         '    '      1, THE POINT IS DRAWN TO FROM THE PREVIOUS POINT
C         '    '      999, THIS SIGNALS THE END OF THE PLOT.
C
      IF (IPEN.EQ.999)  GO TO 10
      IPE = IABS(IPEN-3)
      CALL PLOT(X,Y,IPE)
      RETURN
   10 CALL PLOT(X,Y,IPEN)
      RETURN
      END
```

```
      SUBROUTINE FINISH
C
C  -THIS SUBROUTINE CALLS PROJCT IN ORDER TO PLOT THE
C   UNPLOTTED POINTS REMAINING IN THE BUFFER AFTER ALL
C   POINTS HAVE BEEN STORED OR PLOTTED. IT THEN FLAGS
C   THE END OF THE PLOT FILE BY CALLING SPLOT WITH
C   THE COORDINATES OF THE ORIGIN AND A PEN VALUE OF
C   999.
C
      CALL PROJXN
      CALL SPLOT(0.0,0.0,999)
      RETURN
      END
```

```
      FUNCTION D2R(ANGLE)
C
C  -THIS FUNCTION CONVERTS DEGREES TO RADIANS
C   0.01745329 = PI/180
C
      D2R = ANGLE*0.01745329
      RETURN
      END
```

APPENDIX 2

Explanations of the key

Subroutines

SETUP (VIEW, IVIEW, CENTER, PLTH, PLTW, EYES, RANGE, PFOCAL, ISPHER, SCALE)

SETUP must be the first stereo subroutine called, and should be called only once. It initializes COMMON variables, and sets up vectors used in projection subroutine PROJXN.

Argument	Type	Value	Explanation
VIEW	REAL ARRAY		Location of view point, in either cartesian or spherical coordinates, as set by ISPHER if ISPHER = 0 (cartesian coordinates) - and IVIEW ≠ "VE" VIEW(1),VIEW(2),VIEW(3)= cartesian coordinates of view point - and IVIEW = "VE" VIEW(1),VIEW(2),VIEW(3)= components of view vector (see fig.11b) if ISPHER = 1 (spherical coordinates) - and IVIEW ≠ "VE" VIEW(1),VIEW(2),VIEW(3)= latitude longitude, and elevation of view point. - and IVIEW = "VE" VIEW(1),VIEW(2),VIEW(3)= components of view vector (see fig. 11a)
IVIEW	INTEGER		Flag indicating type of view point
		Blank	VIEW = latitude,longitude,elevation of view point
		"VE"	VIEW = altitude,azimuth,distance of vector from center point
CENTER	REAL ARRAY		Location of center point, in either cartesian or spherical coordinates, as set by ISPHER if ISPHER = 0 (cartesian coordinates) CENTER(1),CENTER(2),CENTER(3) = cartesian coordinates of center point if ISPHER = 1 (spherical coordinates) CENTER(1),CENTER(2),CENTER(3)= spherical coordinates of center point (latitude, longitude,elevation)
PLTH	REAL		Height of plot in inches
PLTW	REAL		Width of plot in inches
EYES	REAL		Eye separation in inches
RANGE	REAL	>0	Depth of field; distance along the view direction between nearest and farthest planes in the figure
		≤0	if RANGE ≤0, depth of field is not taken into consideration

Argument	Type	Value	Explanation
PFOCAL	REAL		Plot focal length, in inches
ISPHER	INTEGER	= 0	VIEW,CENTER, and points to be plotted are input in cartesian coordinates
		= 1	VIEW,CENTER, and points to be plotted are input in spherical coordinates
SCALE	REAL		Scale of plot if ISPHER = 0, scale = cartesian units per inch of plot if ISPHER = 1, scale = kilometers in real world per inch on plot

DRAW3D (P1, P2, P3)

DRAW3D gives the user direct control over pen movement. It is used to draw a continuous line from one point to another in three dimensions.

Argument	Type	Value	Explanation
P1,P2,P3	REAL		Coordinates of point to which the pen draws from previous position. Coordinates may be either spherical or cartesian, as set by ISPHER in initial call to SETUP if ISPHER = 1 (spherical coordinates) P1 = latitude in decimal degrees P2 = longitude in decimal degrees P3 = elevation in kilometers if ISPHER = 0 P1,P2,P3 = cartesian coordinates of point

MOVE3D

MOVE3D gives the user direct control over pen movement. It is used to move from one point to another in three dimensions with the pen up (i.e. without drawing)

Argument	Type	Value	Explanation
P1,P2,P3	REAL		Coordinates of point to which pen moves without drawing Coordinates are either spherical or cartisian, as set by ISPHER in initial call to SETUP if ISPHER = 1 (spherical coordinates) P1 = latitude in decimal degrees P2 = longitude in decimal degrees P3 = elevation in kilometers if ISPHER = 0 P1,P2,P3 = cartesian coordinates of point

JACK (P1,P2,P3, SIZE)

JACK is used to draw a three dimensional "jack" symbol (3 orthogonal line segments intersecting at their midpoints) at the specified location.

Argument	Type	Value	Explanation
P1,P2,P3	REAL		Coordinates of point at which a three dimensional "jack" symbol is drawn. Coordinates may be either spherical or cartesian, as set by ISPHER in call to SETUP if ISPHER = 0 P1,P2,P3 = x,y,z coordinates of point if ISPHER = 1 P1 = latitude in decimal degrees P2 = longitude in decimal degrees P3 = elevation in kilometers
SIZE	REAL		Size in inches of jack in plane of center point. Jacks closer to view point will be larger; jacks farther away will be smaller

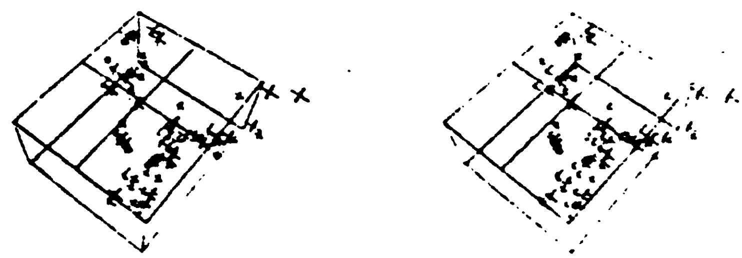
FINISH

FINISH must be called after all data points are input and plotted. This is to complete the plot by processing any points that remain in the buffer. FINISH also flags the end of the output file by calling subroutine SPLOT with the coordinates of the origin and a pen value of "999".

APPENDIX 3

Examples of Input and Output

EXAMPLE 1. Aftershocks of the 1971 San Fernando earthquake, between magnitudes 3.0 and 4.0.



INPUT:

```
VIEW              34.6     -118.6      40.0
CENTER            34.4     -118.4      -8.0
PLOTHEIGHT         3.0
PLOTWIDTH          2.4
EYE SEP            2.4
FOCAL              2.25
SIZE
      3.0  3.5  0.05 0.05
      3.5  4.0  0.10 0.10
      999.
PLOT
     34.309  -118.479  -18.600    3.400     1
     34.341  -118.483   -8.700    3.800     1
     34.346  -118.289  -12.500    3.800     1
     34.354  -118.469   -9.700    3.500     1
     34.314  -118.489  -11.500    3.800     1
     34.352  -118.441  -10.700    3.800     1
     34.415  -118.447  -11.900    3.400     1
     34.432  -118.421  -12.300    3.200     1
     34.332  -118.312   -8.600    3.400     1
     34.428  -118.401   -6.900    3.200     1
     34.441  -118.438  -11.100    3.300     1
     34.373  -118.366   -6.300    3.500     1
     34.353  -118.325   -6.900    3.300     1
     34.417  -118.439   -8.000    3.200     1
     34.293  -118.522  -10.900    3.400     1
     34.445  -118.447  -10.300    3.200     1
     34.457  -118.442  -12.400    3.300     1
     34.397  -118.363   -8.300    3.300     1
     34.337  -118.314   -0.500    3.500     1
     34.296  -118.321   -2.400    3.500     1
     34.374  -118.438  -12.400    3.600     1
     34.392  -118.445  -10.900    3.700     1
     34.448  -118.429  -11.000    3.300     1
     34.352  -118.337   -4.600    3.700     1
     34.357  -118.339   -7.600    3.200     1
     34.412  -118.438   -7.600    3.300     1
     34.339  -118.457   -7.600    3.400     1
     34.407  -118.431   -9.700    3.400     1
     34.411  -118.429   -9.600    3.900     1
     34.319  -118.306   -2.300    3.300     1
     34.374  -118.376    2.000    3.100     1
     34.250  -118.190    0.700    3.100     1
     34.299  -118.511   -1.200    3.800     1
     34.321  -118.290   -4.300    3.400     1
     34.358  -118.466   -7.800    3.200     1
     34.471  -118.411  -12.900    3.900     1
     34.407  -118.441   -7.900    3.400     1
     34.427  -118.435   -9.700    3.400     1
     34.287  -118.544   -0.600    3.500     1
     34.402  -118.453   -9.600    3.300     1
     34.334  -118.298  -10.000    3.100     1
     34.346  -118.492    2.000    3.000     1
     34.338  -118.306   -3.300    3.500     1
     34.383  -118.448   -6.200    3.000     1
```

Example 1 input, continued.

34.347	-118.347	-5.200	3.000	1
34.398	-118.330	-12.100	3.200	1
34.386	-118.429	-4.400	3.200	1
34.374	-118.427	-7.300	3.200	1
34.454	-118.406	-15.200	3.600	1
34.398	-118.426	-4.900	3.500	1
34.398	-118.445	-6.200	3.500	1
34.310	-118.316	-6.400	3.000	1
34.396	-118.328	-2.900	3.500	1
34.441	-119.411	-10.800	3.700	1
34.408	-118.368	-11.100	3.100	1
34.429	-118.453	-12.700	3.200	1
34.336	-118.363	2.000	3.500	1
34.419	-118.383	-7.300	3.500	1
34.451	-118.430	-12.900	3.300	1
34.449	-118.438	-12.000	3.000	1
34.339	-118.286	-5.500	3.000	1
34.376	-118.433	-4.900	3.900	1
34.377	-118.437	-0.800	3.300	1
34.446	-118.448	-2.800	3.000	1
34.468	-118.461	-11.300	3.200	1
34.355	-118.473	-5.700	3.700	1
34.395	-118.429	-7.300	3.500	1
34.284	-118.528	-3.000	4.000	1
34.433	-118.403	-4.800	3.600	1
34.379	-118.431	-7.800	3.000	1
34.277	-118.505	-12.200	3.500	1
34.366	-118.492	-5.100	3.000	1
34.366	-118.492	-5.100	3.000	1
34.3	-118.4	0.0		3
34.5	-118.4	0.0		2
34.5	-118.3	0.0		2
34.3	-118.3	0.0		2
34.3	-118.5	0.0		2
34.5	-118.5	0.0		2
34.5	-118.4	0.0		2
34.4	-118.5	0.0		3
34.4	-118.3	0.0		2
34.3	-118.3	0.0		3
34.3	-118.3	-10.0		2
34.5	-118.3	-10.0		2
34.5	-118.5	-10.0		2
34.3	-118.5	-10.0		2
34.3	-118.3	-10.0		2
34.5	-118.3	0.0		3
34.5	-118.3	-10.0		2
34.3	-118.5	0.0		3
34.5	-118.5	-10.0		2
34.3	-118.5	0.0		3
34.3	-118.5	-10.0		2

APPENDICES

1. Program source code listing
2. Explanation of key subroutines
3. Example of input and output

Figure 1

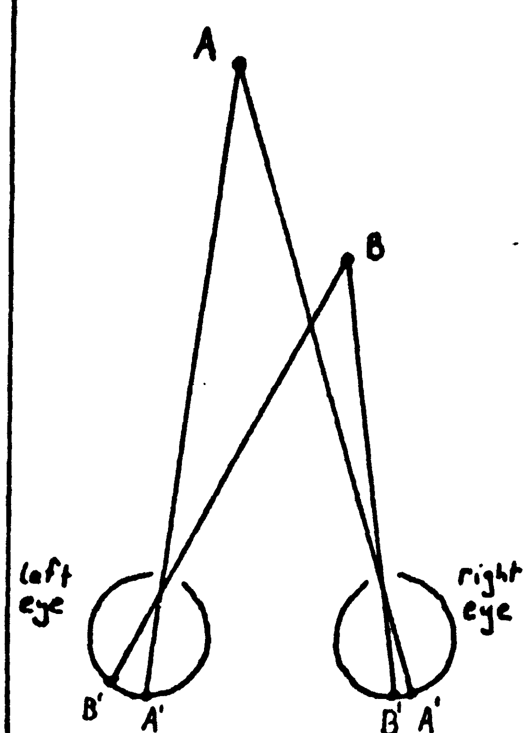


Figure 2

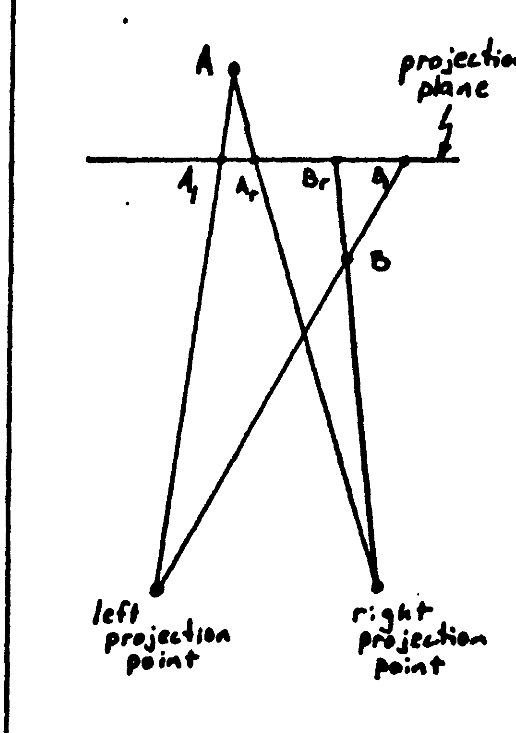


Figure 3

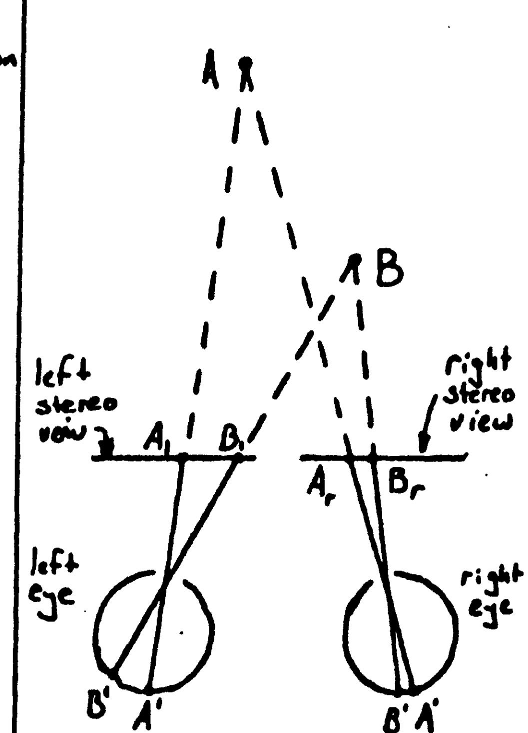


Figure 4

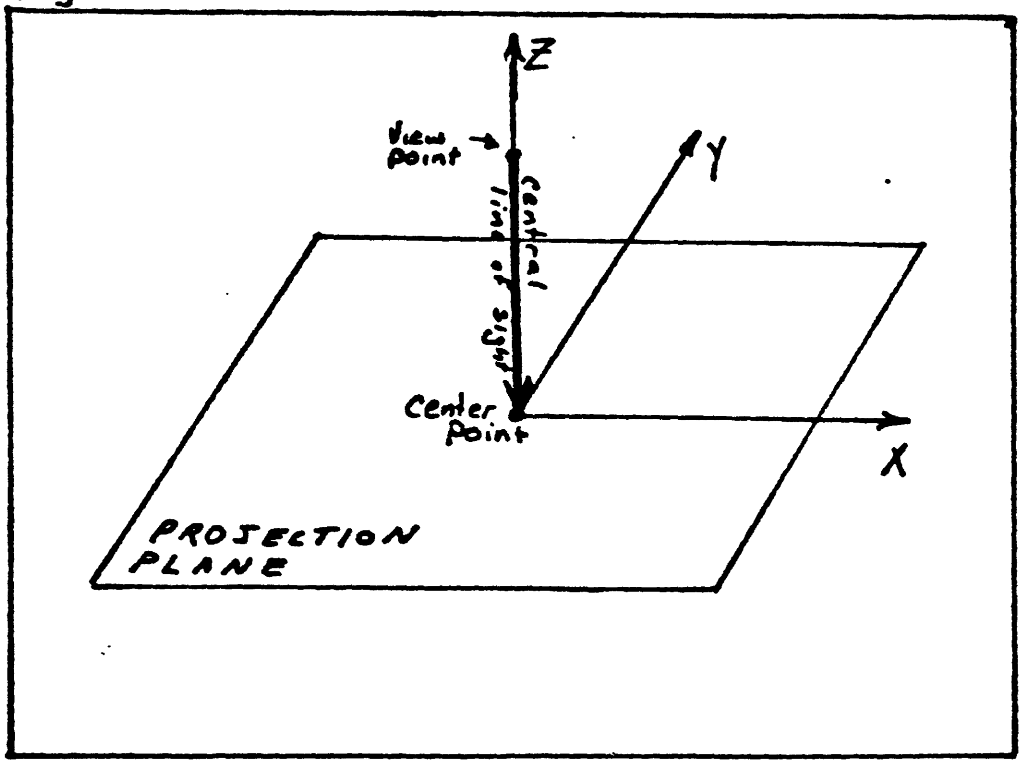
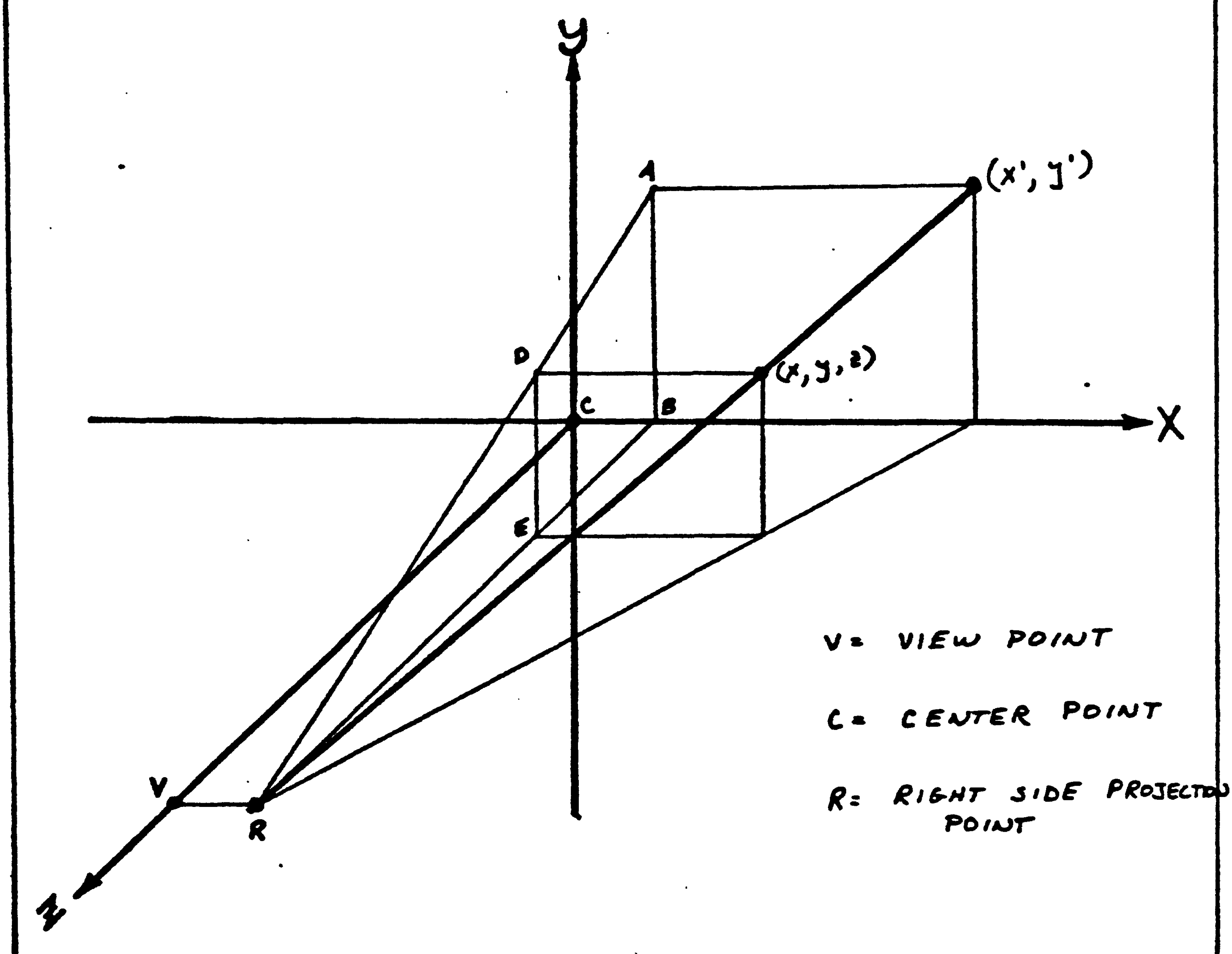

Z
View point
central line of sight
Y
Center Point
X
PROJECTION PLANE

FIGURE 5



y
A
(x', y')
D
(x, y, z)
C
B
X
E
V = VIEW POINT
C = CENTER POINT
R = RIGHT SIDE PROJECTION POINT
V
R
Z

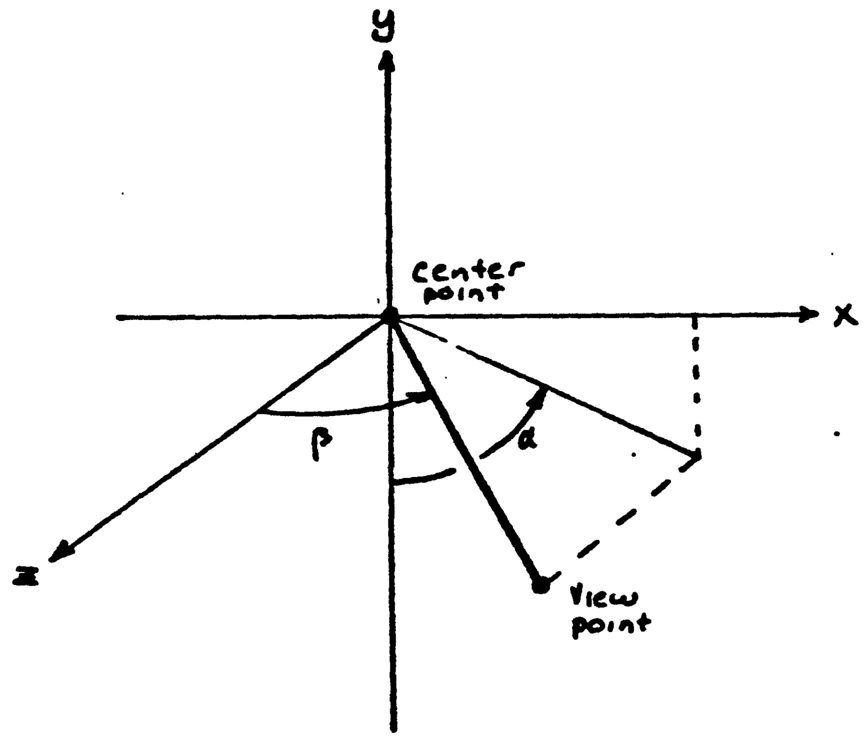
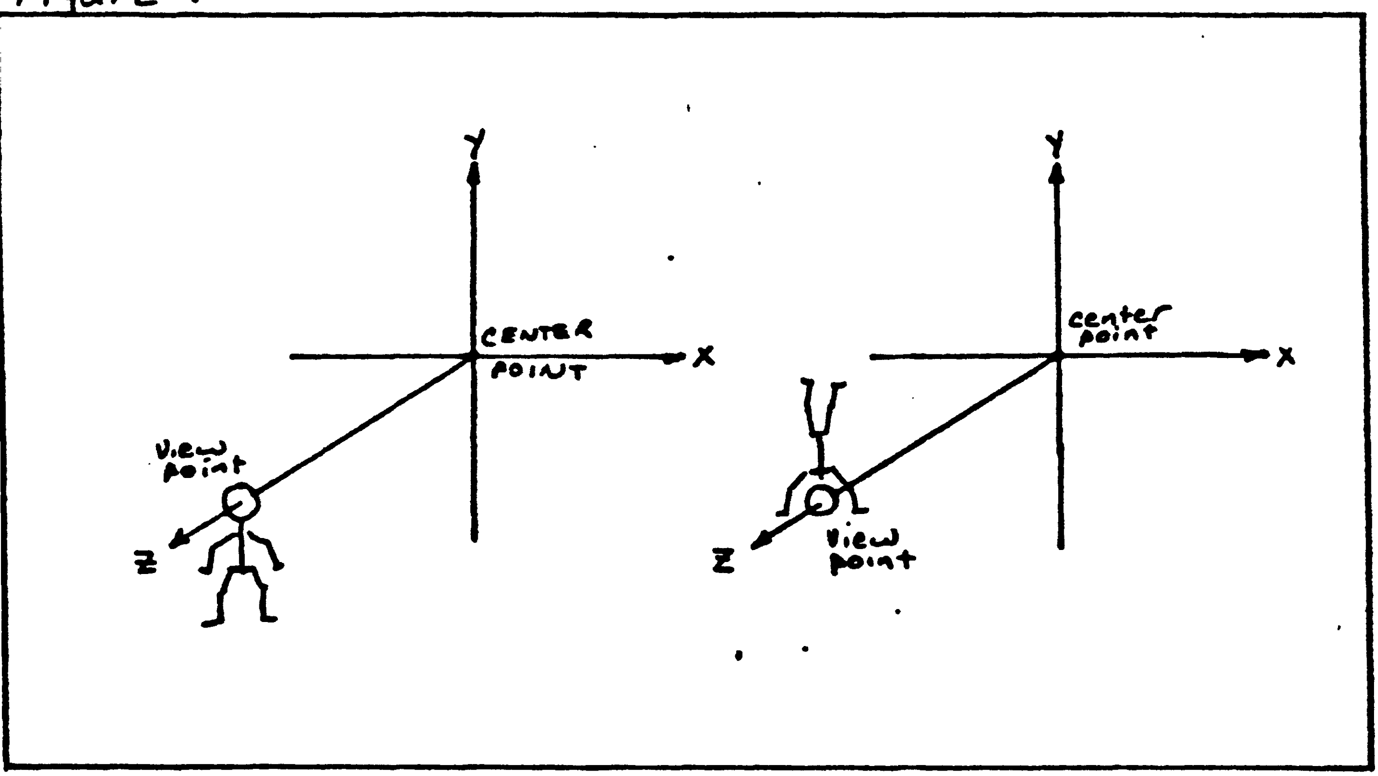


FIGURE 6

Figure 7



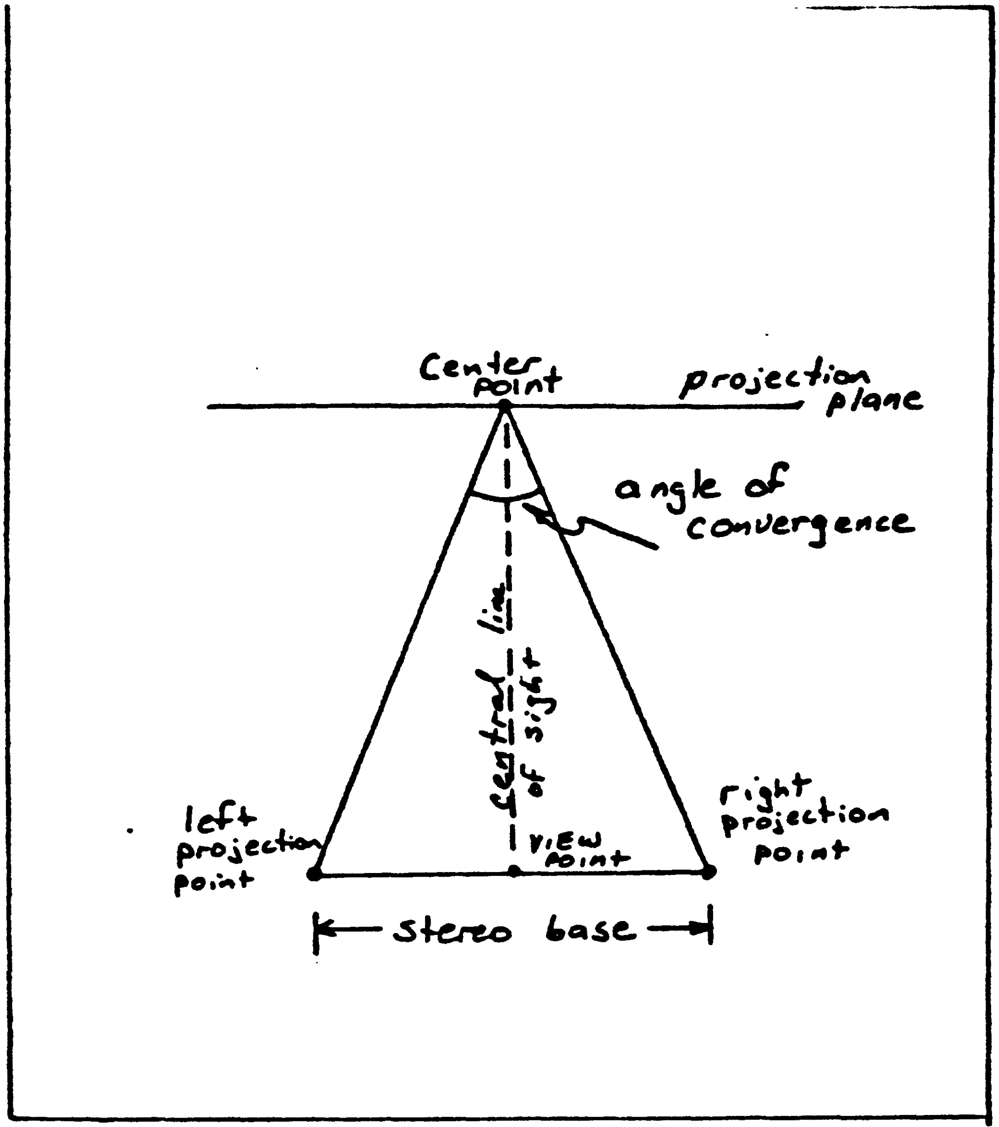
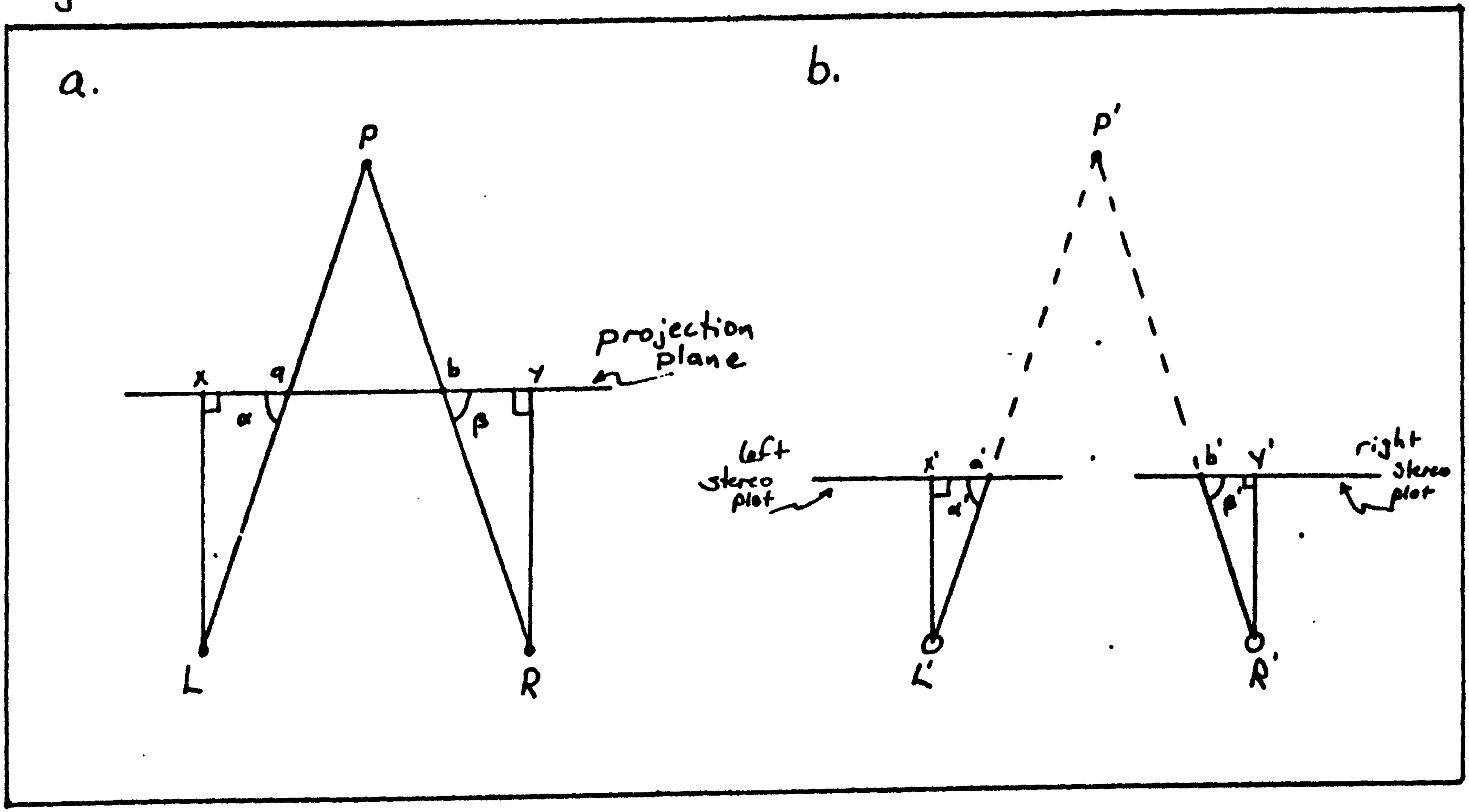

Center point
projection plane
angle of convergence
central line of sight
left projection point
VIEW point
right projection point
stereo base

Figure 8

Figure 9


a.
P
x
a
b
y
α
β
projection plane
L
R
b.
P'
left stereo plot
x'
a'
α'
b'
y'
β'
right stereo plot
L'
R'

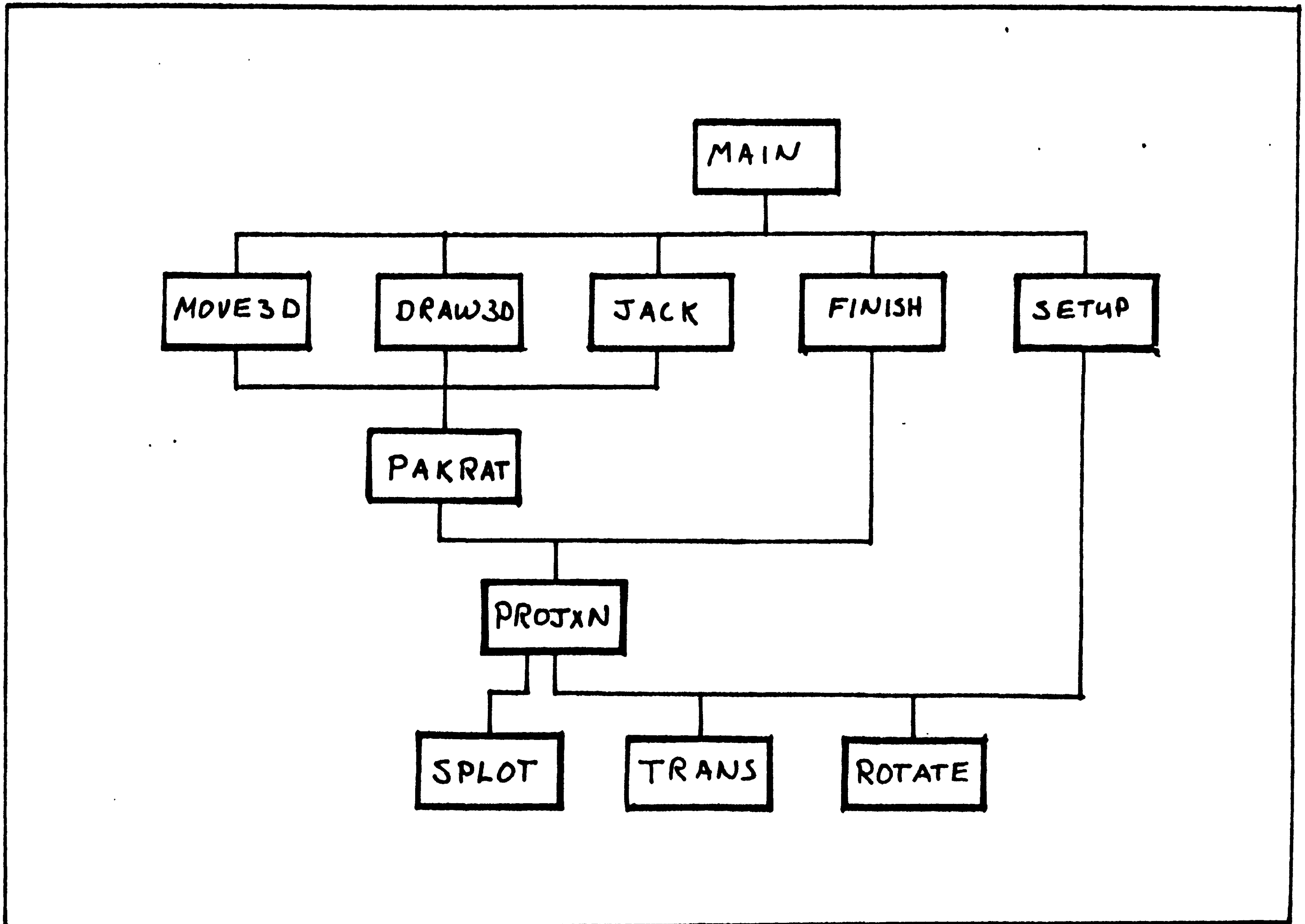
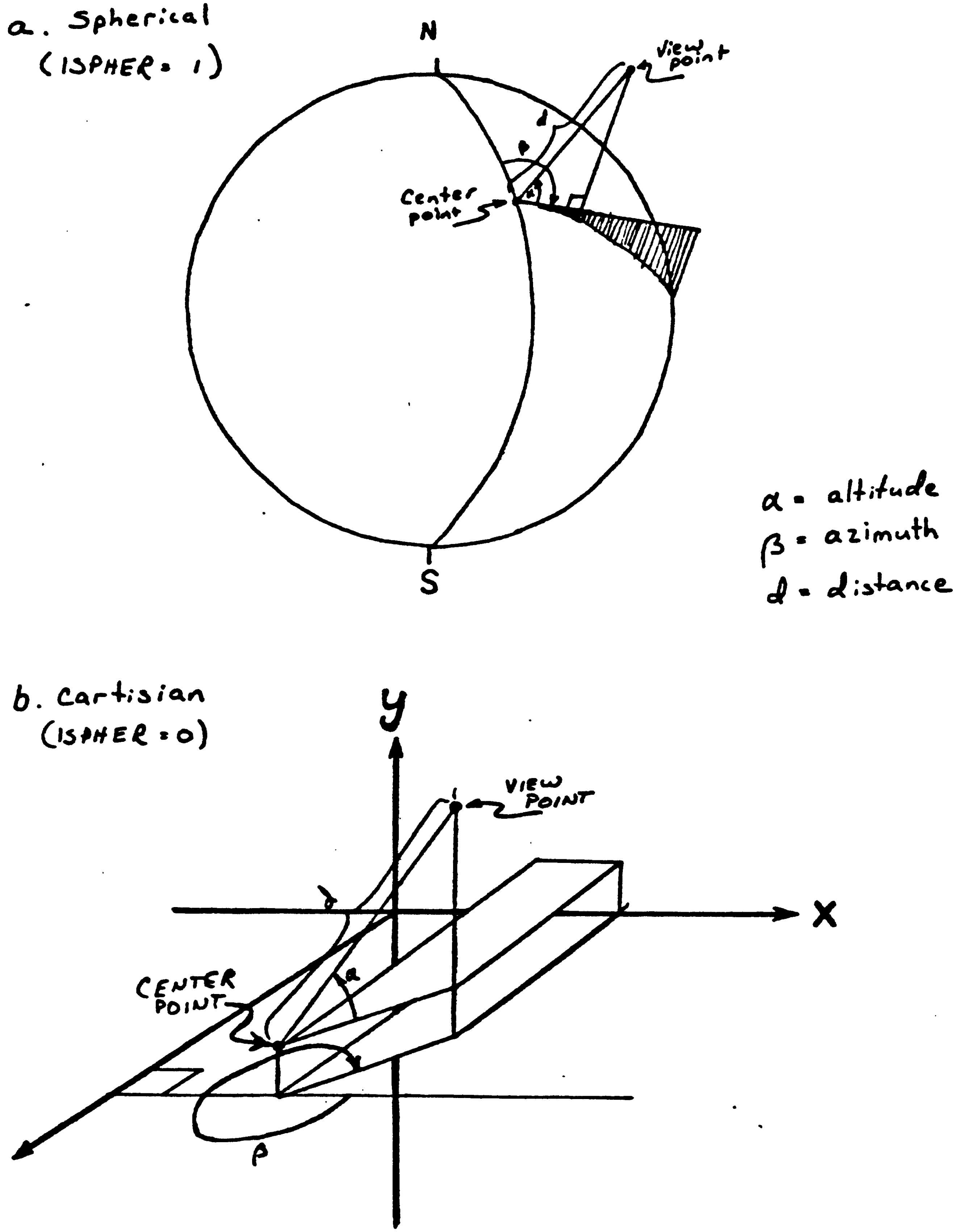

MAIN
MOVE3D
DRAW3D
JACK
FINISH
SETUP
PAKRAT
PROJXN
SPLOT
TRANS
ROTATE

Figure 10

FIGURE 11


a. Spherical
(ISPHER = 1)
N
view point
d
Center point
S
α = altitude
β = azimuth
d = distance
b. Cartisian
(ISPHER = 0)
y
VIEW POINT
X
CENTER POINT
α
β
Z