A GRAPHICS PACKAGE FOR SIMULATING THE DATA OF RANGE FINDERS

STEPHEN H.Y. HUNG

National Research Council of Canada
Ottawa, Ontario, Canada K1A 0R8

Abstract: A graphics package for generating the range data of 3-D objects is described. The simulating data generated by this package can be used for 3-D object recognition or related research. It is a useful tool for someone who wants to work on range data of 3-D objects, but does not have a good range finder.

Key Words: 3-D range data, Geometrical modeling, Range finder type scanner, Simulating data.

I. Introduction

In our recent study, we needed a very large amount of range data to represent the images of objects from all possible viewing positions. A laser range finder type scanner [1] was used to obtain some of the needed data. However, there were two problems: firstly, the amount of data needed was too large and it was too difficult to set up an object at an angle that is accurate enough for our purpose with the equipment available to us; secondly, the noise and the possible errors of the scanner blurred the real picture of the behavior of the images. At this stage, it was difficult for us to determine whether certain behavior is a real character of the images or if it is caused by the noise or the error of the scanners. Thus, we have decided to use a graphics modeling package to generate simulating data for the preliminary investigation.

There are many geometrical modeling packages available, such as UNIBLOCK , ROMULUS, PROREN and COMPAC [2]. However, most of these are too general or too complicated for our purpose on most aspects, but are not accurate enough on some particular requirements. We have decided to write a package especially for our own purpose. Such a package will be quite simple and will have only the exact features that are needed. In the future, if so desired, this package can easily be included in the overall system to facilitate the recognition processing. It certainly will be a useful tool for someone who wants to work on range data of 3-D objects, but does not have a good range finder.

In the following sections, we will describe the features of this package and the principles on which it is based. This package has been used to generate 3-D range data for our study on the choice of features for characterizing the images of an object from all possible viewing positions, and the possibility of establishing a feasible recognition system based on this approach.

II. The Basic Assumptions

The range finder we are using is a laser range finder type scanner, described in ref. 1, which provides a two-dimensional image of Z(x,y) from a zero reference plane (x,y) orthogonal to the line of sight from the scanner. The graphics package will act in a manner similar to this scanner. We are assuming that:

1. The readings of X, Y and Z will be given in millimetres according to a fixed coordinate system.

2. A reading of Z will be taken at every 0.5 mm interval along both X and Y axes.

3. When transformations of rotations are applied to an object, the object always rotates along the X axis first and then rotates along the Y axis, unless specified otherwise.

4. In each viewing position, the object will be generated with the lowest point (may or may not be visible) just touching the reference plane and the image of the range data thus obtained is considered as the standard of this particular viewing position. Of course, a translation along the Z axis can be applied to move the object up or down as desired.

5. We are dealing only with an object that can be described by the combination of its surfaces, and each surface can be seen as one region or divided into two or more regions such that each region can be described in the X(s,t), Y(s,t) and Z(s,t) Bezier form of parametric equations.

6. The position of the scanner is fixed. Different viewing angles are obtained by rotating the object along the X and Y axes.

7. The particular scanner we are going to use will scan from 1000 mm above the reference plane for an area about 128 mm × 128 mm, i.e., 256 × 256 points.

III. The Descriptions of Surfaces

An object is described by all its surfaces. A surface can either be seen as a region or can be divided into regions such that each region can be described in one of the following ways:

1. A region is bounded by four straight lines and determined by four points. The surface in Fig. 1a is determined by points 1, 2, 3 and 4 and can be described as {1,2,3,4} (but not {1,2,4,3}, Fig. 1b). This includes the cases where a region is bounded by three lines ({1,2,3,2} in Fig. 1c).

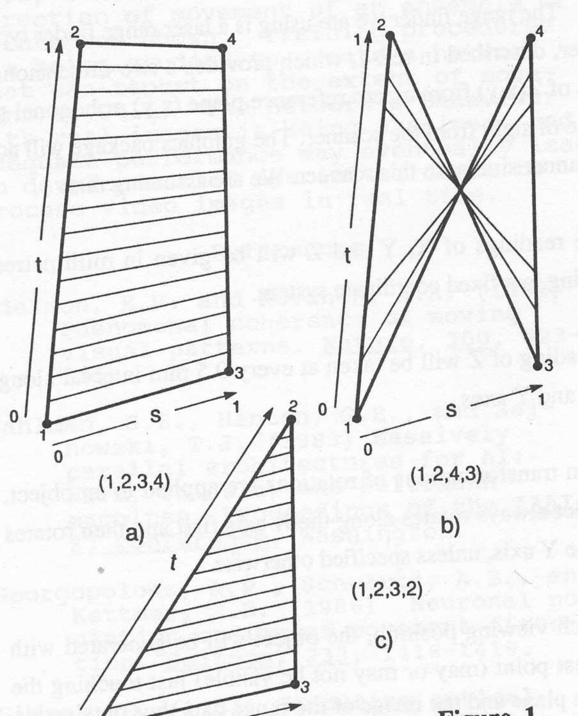

Figure 1

2. A region is bounded by two lines and two curves, and determined by eight points as in Fig. 2a and 2b. The special cases are shown in Fig. 2c and 2d.

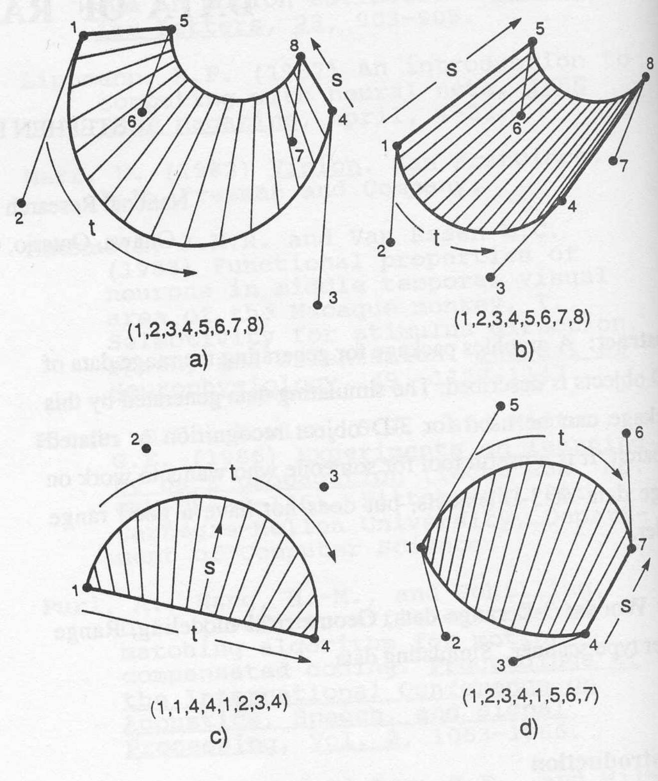

Figure 2

3. Regions that do not belong to 1 or 2 will be determined by 16 key points as in Fig. 3a. A possible special case is shown in 3b.

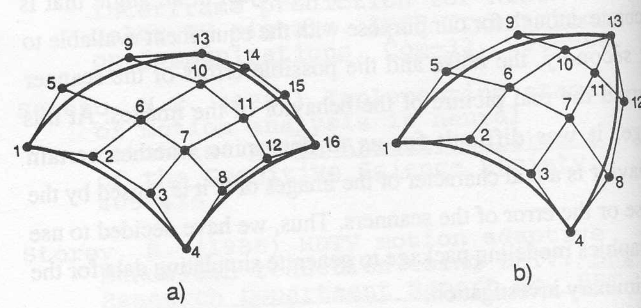

Figure 3

Remark 1: The four key points in case 1 do not have to be in the same plane. The regions in case 2 could be flat or a part of the surface of a cylinder or a cone (the generator is a straight line). A flat region in case 3 (a flat region bounded by four curves) can always be reduced to a special case in case 2.

IV. Data Generation

For each region, the maximum and minimum values of X and Y among all its key points are first obtained and are then used to determine the boundary of the area that must be concerned (see Fig. 4). Then the differences between the maximums and minimums are also computed as:

$$DX = X_{max} - X_{min}, \qquad DY = Y_{max} - Y_{min}$$

if $DX \geq DY$, then we process along the X axis. The process starts at X_{min} with an interval of 0.5 mm, until X_{max}. For each X value, Y starts at Y_{min} with a 0.5 mm interval until Y_{max}.

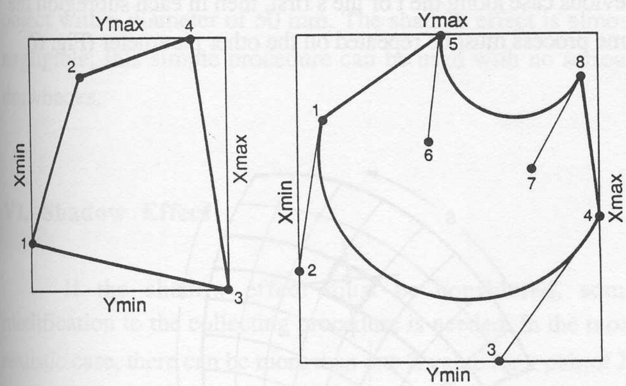

Figure 4

(1). For the regions of the first kind, each pair of X and Y can be used to solve the equations:

$$X(s,t) = s \cdot t \cdot SUX_4 + s \cdot SUX_3 + t \cdot SUX_2 + SUX_1$$

$$Y(s,t) = s \cdot t \cdot SUY_4 + s \cdot SUY_3 + t \cdot SUY_2 + SUY_1$$

where:

$$SUX_1 = PX_1$$

$$SUX_2 = PX_2 - PX_1$$

$$SUX_3 = PX_3 - PX_1$$

$$SUX_4 = PX_4 - PX_3 - PX_2 + PX_1$$

PX_i, i = 1, 4 is the X coordinate of key point i

(similar for SUY_i and PY_i)

and get a pair of (s,t). If the s or the t is greater than 1 or less than 0, it is dropped. Otherwise, we subsititute it in the equation:

$$Z(s,t) = s \cdot t \cdot SUZ_4 + s \cdot SUZ_3 + t \cdot SUZ_2 + SUZ_1$$

(similar for SUZ_i and PZ_i as for X)

and get the Z value for the point (X,Y).

(2). For the regions in 2, the parametric equations are:

$$X(s,t) = s \cdot t^3 \cdot SUX_8 + s \cdot t^2 \cdot SUX_7 + s \cdot t \cdot SUX_6 + s \cdot SUX_5 + t^3 \cdot SUX_4 + t^2 \cdot SUX_3 + t \cdot SUX_2 + SUX_1$$

where:

$$SUX_1 = PX_1$$

$$SUX_2 = 3 \cdot (PX_2 - PX_1)$$

$$SUX_3 = 3 \cdot (PX_1 - 2 \cdot PX_2 + PX_3)$$

$$SUX_4 = PX_4 - PX_1 + 3 \cdot (PX_2 - PX_3)$$

$$SUX_5 = PX_5 - PX_1$$

$$SUX_6 = 3 \cdot (PX_1 - PX_2 - PX_5 + PX_6)$$

$$SUX_7 = 3 \cdot (PX_5 - 2 \cdot PX_6 + PX_7 - PX_1 + 2 \cdot PX_2 - PX_3)$$

$$SUX_8 = PX_1 - PX_4 - PX_5 + PX_8 + 3 \cdot (PX_6 - PX_7 - PX_2 + PX_3)$$

All these are similar for Y(s,t) and Z(s,t) .

Sine the parametric equations contain the cubic of t, it will be difficult to solve the (s,t) directly as in case 1 when given X and Y. To avoid this difficulty, the following procedure is used:

A. Let $\delta T = 0.5/\omega_{xyz}$, where $\omega_{xyz} = MAX\{DX,DY,DZ\}$. DZ is defined in the same way as DX and DY described above.

B. $t_1 = 0.0$, $t_2 = t_1 + \delta T$, ..., $t_i = t_{i-1} + \delta T$ until there is an 'n' such that $t_n \geq 1.0$, then let $t_n = 1.0$.

C. The curves C(0,t) and C(1,t) are each divided into n-1 sections with t_i, i=1,n as the dividing points. Thus, the region is divided into n-1 subregions by connecting the corresponding dividing points in C(0,t) and C(1,t) with the line $L(s,t_i)$, i=1,n. Such subregions have the same parametric equations as the whole region but the t is restricted between t_i and t_{i+1} for subregion i.

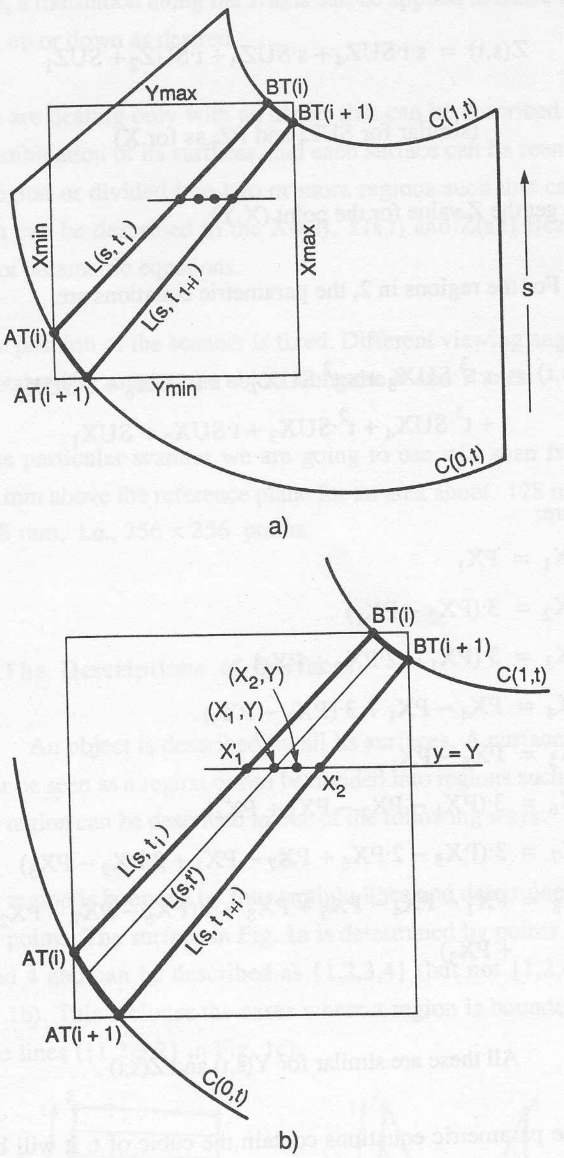

Figure 5

D. The subregions are processed one by one from 1 to n-1. For subregion i, as indicated in Fig. 5, the DX and DY are found from the four points, AT(i), AT(i+1), BT(i) and BT(i+1), as described previously. Assuming $DY \geq DX$, we proceed along the Y axis. For a Y between Y_{max} and Y_{min}, the plane y = Y will intersect the lines $L(s,t_i)$ and $L(s,t_{i+1})$ at the points with the X coordinates X'(i) and X'(i+1), respectively. The X values between X'(i) and X'(i+1) that are also the multiples of 0.5 mm are the sampling points at this Y value; for example, we have (X_1,Y) and (X_2,Y).

E. In order to find the s and t for (X_1,Y), we use the proportion of $(X_1, X'(i))$ and $(X_1,X'(i+1))$ as a clue to take a value t' between t_i and t_{i+1}. If the line L(s,t') intersects y = Y at a point with X' such that $ABS(X'-X_1) \leq 0.0001$ (or any preset limit), then the t' will taken as the t for (X_1,Y). Otherwise, the same process will be repeated between t' and t_i or t_{i+1}, depending on the result of the previous iteration, until the preset limit is satisfied.

F. Once the t' is fixed, the s can be obtained easily. After making sure t' is between t_i and t_{i+1}, and s is between 0.0 and 1.0, they can be substituted into the equation Z(s,t) to compute the Z value for (X_1,Y). The procedure is similar for (X_2,Y).

(3). The parametric equations for the most general cases of the regions in case 3 are too complicated to be included here, but can be found in any computer graphics book such as Ref. 3. Since both s and t have cubics, they must be treated as in the previous case along the t or the s first, then in each subregion the same process must be repeated on the other parameter (Fig. 6).

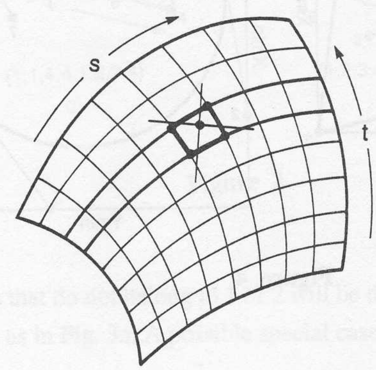

Figure 6

V. Data Collection and Hidden Part Removal

When the data of a region are generated, they will be embedded, point by point, into the data set of the object. The data of the object are arranged along the X values first, from the smaller one to the larger one, and within the same X value, from smaller Y to larger Y.

For each point of the newly generated region, firstly, its X value is compared with the X-index rather than to each point in the data set of the object. In our particular case, there are at the most 256 values in the X-index. When the X value of a point is compared to the X index:

A. If there is no such value, then this is considered to be a new point and it will be recorded in the data of the object; at the same time, the X value will be added to the X-index.

B. If there is such an X, then the Y value of the point under processing will be compared with the Y value of each point with the same X value in the object data set. If there is no such Y value, then this is a new point and it will be recorded in the object data set; if there is such a Y value, then the Z values of these two points, which have the same X and Y values, will be compared and only the one with the larger Z value will be recorded.

This embedding process also removes the hidden part if the viewing position is assumed to be infinity. In our particular case the scanner is scanning, from a distance of 1000 mm, an object with a diameter of 50 mm. The shadow effect is almost negligible; this simple procedure can be used with no serious drawbacks.

VI. Shadow Effect

If the shadow effect must be considered, some modification to the collecting procedure is needed. In the most realistic case, there can be more than one Z value for a pair of X and Y (Fig. 7a). However, we will consider the quasi-realistic cases in which only one Z value is allowed for each pair of X and Y. For some points, even the highest Z value still must be dropped because of the shadow effect, as in Fig. 7b. The same procedure can be extended to the most realistic cases, but must be accompanied by some change in the generation of data for vertical surfaces.

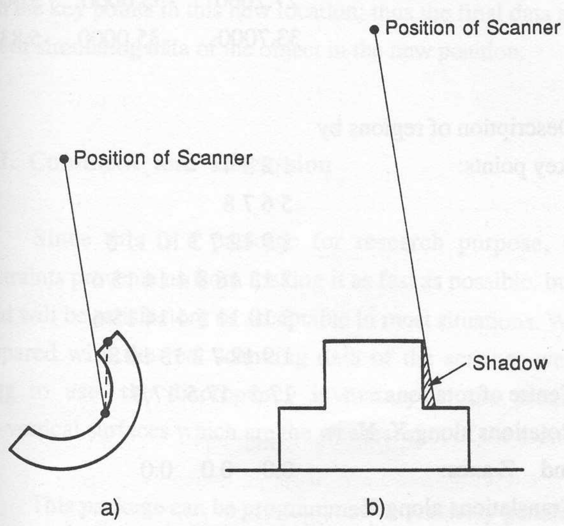

Figure 7

In the quasi-realistic cases, the testing for shadow effect will begin after the collection of the object data set is finished. We can assume that the highest value z' of Z of the object is known.

When a point is being tested, if its Z value is equal to the highest Z value z' of the object, then it cannot be in a shadow. Otherwise, this point (x_0, y_0, z_0) and the point (x_s, y_s, z_s) of the position of the scanner will be connected by a straight line with the parametric equations:

$$X(p) = (1-p)\cdot x_0 + p\cdot x_s$$

$$Y(p) = (1-p)\cdot y_0 + p\cdot y_s$$

$$Z(p) = (1-p)\cdot z_0 + p\cdot z_s$$

Since the highest Z value is z', the portion of the straight line higher than z' will never intersect any surface of this object, thus only the lower portion of the line must be considered, i.e., the portion where $0.0 \le p \le p'$, where p' is a value of p such that $Z(p') = z'$.

By substituting p' into the other two equations, we will get the $x'=X(p')$ and $y'=Y(p')$. All we have to worry about is the segment determined by (x',y',z') and (x_0,y_0,z_0). Supposing that the projection of this segment in the (x,y) plane is a straight segment determined by (x',y') and (x_0, y_0), let it be called LM. The area consisting of all the squares intersecting LM is called the frontier (as indicated by shadow in Fig. 8). Any data point located at any corner (grid point) of those squares will be considered. One should note that:

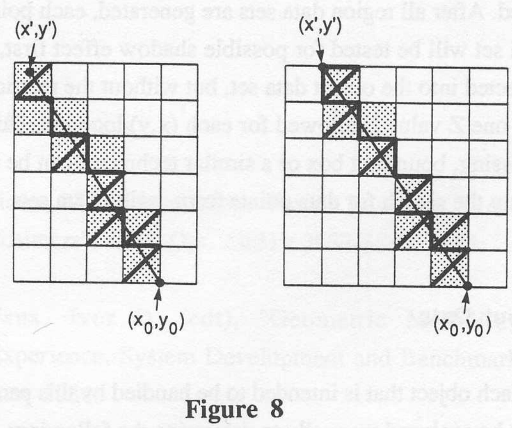

Figure 8

a. While the (x_0, y_0) is always at a grid point of the coordinate, the (x',y') can appear anywhere between.

b. There may or may not be a data point located at each of those locations that we are concerned about. If there is a data point at a particular location, then it should be connected to all other data points of the same square; the connecting segment is called CS. The point (x_0 ,y_0) will never be connected. However, if the point (x',y') is located at one of the grid points, then it should also be connected to other data points of the same square.

c. If a CS does not intersect LM, then it will be ignored. Otherwise, it will be tested.

The test that will be applied to all the remaining jointing segments can be described as follows:

1. For a connecting segment CS, the X and Y coordinates of the intersection of CS and LM will be computed first.

2. The Z value Z(lm) of LM at the intersection point will be computed from the parametric equations given above.

3. The Z value Z(cs) of the connecting segment CS will be computed from its two endpoints by linear interpolation.

4. If Z(lm) > Z(cs), then we say the test result is an negative one. Otherwise, i.e., Z(lm) ≤ Z(cs), it is positive.

If at any step a positive result is obtained, then the point under testing will be dropped, i.e., it is in the shadow of some region. If all the test results are negative, then the point will be kept in the final data set.

In the most realistic case, each region data set will not be combined into a big object data set before the shadow effect is considered. After all region data sets are generated, each point in each data set will be tested for possible shadow effect first, and then collected into the object data set, but without the restriction that only one Z value is allowed for each (x,y) location. During the processing, bounding box or a similar technique can be used to speed up the search for data points from region data sets.

VII. Input Data

Each object that is intended to be handled by this package must first be analyzed manually to determine the following:

A. Each surface of the object, can either be seen as a region or be divided into two or more regions such that each region can be determined by four points, if it is bounded by four straight lines; by eight points, if it is bounded by two lines and two curves; and 16 points, otherwise.

B. The coordinates of all points which determine the regions (called key points) according to a chosen coordinate system.

The actual input file will best be described through an example. The object in Fig. 9 consists of six surfaces and 16 key points. Among those surfaces, S1 and S2 are determined by four key points; the rest are determined by eight points. The input data file for this object is itemized as follows:

1. Position of the scanner: 17.5 -17.5 1000.0

2. Number of key points, and number of regions of '1', '2' and '3': 16 2 4 0

3. Coordinates of key points (in X, Y, Z form):

0.0000	0.0000	17.5000
0.0000	-35.0000	17.5000
7.0000	0.0000	17.5000
7.0000	-35.0000	17.5000
28.0000	0.0000	17.5000
28.0000	-35.0000	17.5000
35.0000	0.0000	17.5000
35.0000	-35.0000	17.5000
1.3000	0.0000	-5.8334
7.8000	0.0000	3.5000
27.2000	0.0000	3.5000
33.7000	0.0000	-5.8334
1.3000	-35.0000	-5.8334
7.8000	-35.0000	3.5000
27.2000	-35.0000	3.5000
33.7000	-35.0000	-5.8334

4. Description of regions by key points:

1 2 3 4
5 6 7 8
1 9 12 7 3 10 11 5
2 13 16 8 4 14 15 6
3 10 11 5 4 14 15 6
1 9 12 7 2 13 16 8

5. Centre of rotations : 17.5 -17.5 17.5

6. Rotations along X, Y and Z axes: 0.0 0.0 0.0

7. Translations along X, Y and Z axes: 0.0 0.0 0.0

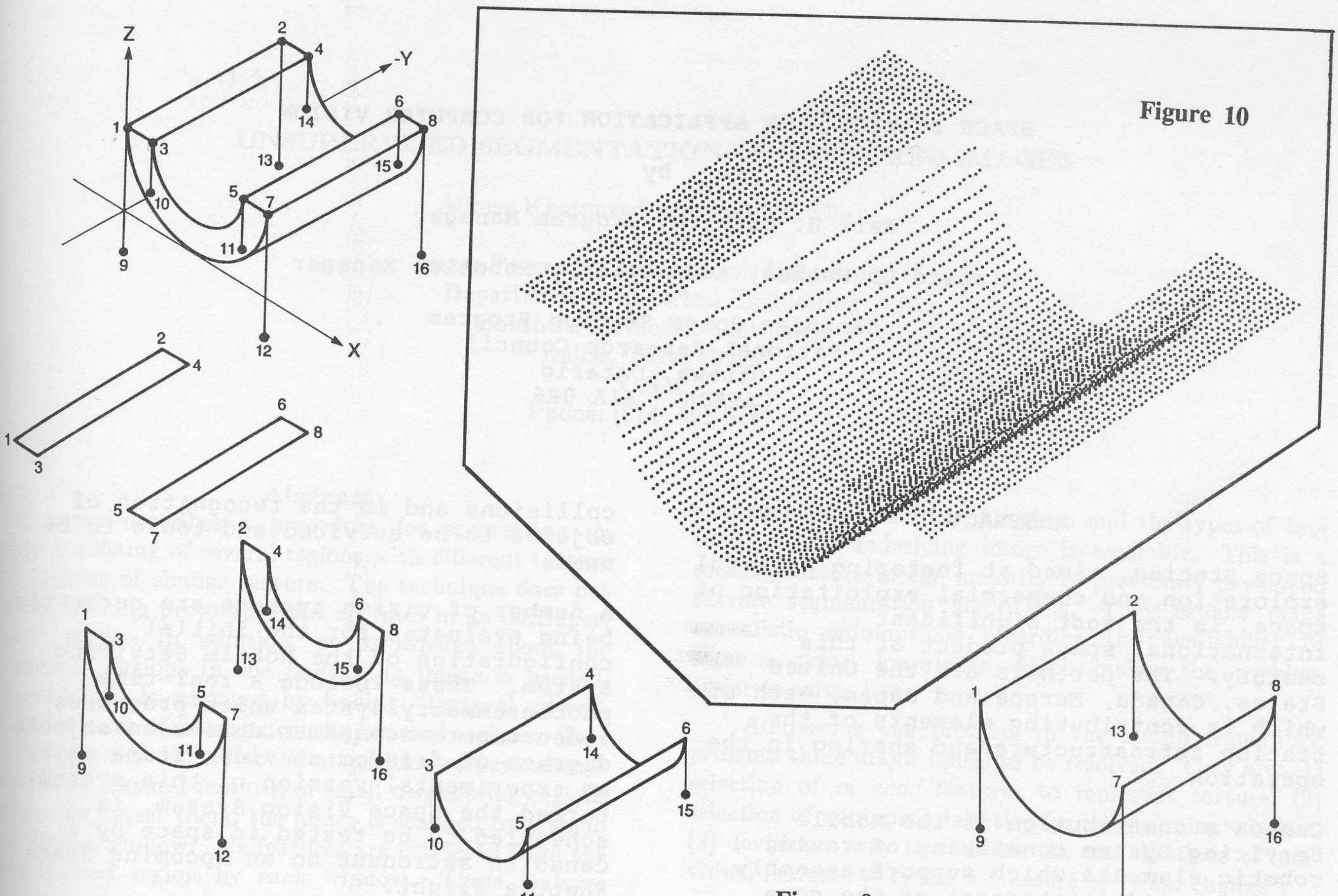

Figure 9

Items 5, 6 and 7 can be repeated as many times as desired or can be programmed to vary in any way. The file described above will generated the object in the standard position, as in Fig. 9. Figure 10 is the plot of the simulating data generated from the set of input data given above.

Remark 2: The rotations and translations are applied to the key points. Once in a new location, each region will be generated from the key points in this new location; thus the final data set is a set of simulating data of the object in the new position.

VIII. Comment and Discussion

Since this is a package for research purpose, time constraints prevents us from making it as fast as possible, but its speed will be satisfactory or acceptable in most situations. When compared with the real scanning data of the scanner we are going to use, the discrepancy is mainly in the areas of near-vertical surfaces which are the weakest spot of the scanner.

This package can be programmed to precisely generate a series of data sets of an object at any rate of increment of rotating angles or translations. It is indeed a useful tool for studying the characteristics of 3-D objects, especially when a range finder is not available. It can also be included in a recognition system to facilitate some procedures, such as the testing of the accuracy of a set of affine transformations that were found by applying the found affine transformation to the key points of an object, and then comparing the data generated at the new position with the observed data.

IX. References

1. Rioux, M., "Laser Range Finder Based on Synchronized Scanners". Appl. Opt. 23(21): 3837-3844; 1984.

2. Faux, Ivor D. (edt), "Geometric Modelling, User Experience, System Development and Benchmark Tests". June 1984.

3. Foley, J.D., Van Dam, A., "Fundamentals of Interactive Computer Graphics". Addison-Wesley Publishing Company, 1982.