Range Data Processing: Representation of Surfaces by Edges[1]

B. Bhanu, S.K. Lee, C.C. Ho and T.C. Henderson
UUCS 85-114
November 27, 1985

Department of Computer Science
University of Utah
Salt Lake City, Utah 84112

Abstract

Representation of surfaces by edges is an important and integral part of a robust 3-D model based recognition scheme. Edges in a range image describe the intrinsic characteristics about the shape of the objects. In this paper we present three approaches for detecting edges in 3-D range data. The approaches are based on computing the gradient, thresholding, thinning and fitting straight lines or curves; fitting 3-D lines to a set of points, and detecting changes in the direction of unit normal vectors on the surface. These approaches are applied locally in a small neighborhood of a point. The neighbors of a 3-D point are found by using the k-d tree algorithm. As compared to previous work on range processing, the approaches presented here are applicable not only to sensor range data corresponding to any one view of the scene, but also to 3-D model data obtained using the Computer-Aided Geometric Design (CAGD) techniques, and to 3-D model built using the sensor data such as the data obtained by combining several views of an object. We present several examples where the data is synthetically genetrated, obtained from CAGD methods or obtained from a laser scanner. A comparison of the techniques is presnted.

Index Terms: 3-D Models, 3-D shape, Computer-Aided Design, Computer-Aided Geometric Design, Edge Detection, Intrinsic Surface Characteristics, Range Data Processing

Name and Present Address of the Corresponding author:

Bir Bhanu, Department of Computer Science, 3160 Merrill Engg. Building, University of Utah, Salt Lake City, Utah 84112

---

[1]This work was supported in part by NSF Grants DCR-8506393, DMC-8502115, ECS-8307483 and MCS-8221750

1. Introduction

An efficient and invariant representation of surface of 3-D objects is of fundamental importance in model based recognition of 3-D objects, and their manipulation by robots [1, 3, 5, 6]. In the field of computer-aided geometric design (CAGD) surface/boundary representations include Coons patches, bicubic surface patches, Bezier methods and B-splines [2]. In our work related to CAGD based 3-D vision [6, 7], we are investigating several approaches which allow the generation of computer representations and geometric and functional models of complicated realizable 3-D objects in a *systematic manner*. The approaches to generate vision models from CAGD models are [7]: universal representation of objects by surface points, intrinsic surface characteristics defined by points of high curvature/edges/arcs, and higher level surface description. Some of these representations like the curvature and surface description can be directly obtained from the design. In this paper we present three approaches for extracting edges in the range data. Although in general, the edge detection techniques have the drawback in that the edge responses must be grouped, thinned and linked in order to produce a reasonable object description in terms of coherent regions, they have the additional feature in that once the line segments are found, the theory of 3-D line semantics can be directly applied and edges can be very useful in 3-D shape recognition [9, 12].

The approaches presented in this paper are applicable to the range data such as the range images obtained using a commercially available laser scanner, to 3-D model data obtained from CAGD, and to 3-D model built using the sensor data [5]. The approaches are based on computing the gradient, thresholding, thinning and fitting straight lines or curves; fitting 3-D lines to a set of points, and detecting changes in the direction of unit normal vectors on the surface. These approaches are applied locally in a small neighborhood of a point. The neighbors of a 3-D point are found by using the k-d tree algorithm. We present several examples where the data is synthetically generated, obtained from a laser scanner or CAGD methods. A comparison of the techniques is presented.

2. Edge Detection in Range Images

In range images edge points correspond to those points which lie on significantly different regions of the surface. Like the edge detection in intensity images, a number of techniques have been used for the segmentation of range images [13, 14, 15, 17, 19]. An excellent survey of these techniques is given in [3]. The above techniques have been applied to one view of the range data in the form of an image [13, 14, 17]; they have not been applied to 3-D model data. The 3-D edge detector of Zucker and Hummel [19] is not directly applicable to find the edges in a surface based representation where only the (x,y,z) coordinate points are available on the surface of an object. The 3-D model data can be acquired from CAGD or from the processing of sensor data [5, 16].

3. Algorithms for Surface Characterization by Edges

In the next section we present three algorithms for edge detection which can be used on both the single view range image and the 3-D surface model.

Technique A. Gradient Approach:

A second derivative formulation for discrete case is used to calculate magnitude and direction of edge at each point similar to the work by Sugihara [17]. Given the range value r at a point (i,j), magnitude and direction of an edge is given by the following equations.

```
m(i,j; 0) = [r(i,j-k)+r(i,j+k)-2*r(i,j)]/(2*k),

m(i,j; 45) = [r(i-k,j+k)+r(i+k,j-k)-2*r(i,j)]/(2*sqrt(2)*k),

m(i,j; 90) = [r(i-k,j)+r(i+k,j)-2*r(i,j)]/(2*k),
```

$$m(i,j;\ 135) = [r(i-k,j-k)+r(i+k,j+k)-2*d(i,j)]/(2*sqrt(2)*k),$$

It can be seen that the values of these operations are zeroes on flat surfaces, positive on convex edges and negative on concave edges, and m(i,j; *theta*) is most sensitive to the edge which lies in the direction of *theta*. The maximum value of m(i,j; *theta*) is the maximum magnitude of gradient and *theta* is the direction of edge at this point.

The simplest technique for thinning is to retain only those edge points whose magnitude is a local maximum based on edge direction. An edge element is said to be present at a point if:

1. The edge magnitude of the central point exceeds some threshold value.
2. The edge magnitude at the point is larger than the its two neighbors in a direction normal to the direction of this edge when the directions of the two neighboring points are the same of that of the central point.

To link the edge points, we used the edge magnitude and the edge direction together. We find the candidate points which can be linked to the central point smoothly. And if there are more than two candidates, we link the point which has larger magnitude than other candidates.

Note that the technique assumes that the range data is in the form of an image. In this context it is similar in concept to the work reported in [13].

Technique B. Line Fitting Approach:

In this approach we do not assume that the data is in the form of an image. Normally it is in the form of a list of (x,y,z) points and it can be *nonuniformly spaced*. The neighbors of a point are found by using the k-d (k=3) tree algorithm to organize the data [11]. A k-d tree is a binary tree of k-dimensional keys which is organized such that at each subdivision step, the data are split at the median along the axis having the greatest spread in vector element along that axis. The data can be organized in O(nlog n) time and it allows the determination of m-nearest neighbors of a given query in O(log n).

After getting neighboring points in 3-D space, we calculate the unit direction vector from each point to its neighboring points. The two direction vectors lie on a straight line if these two direction vectors point in exactly opposite directions. If we find two or more than two straight lines within a certain threshold (related to the differences of the direction vectors), then this means that center point and all neighboring points lie on a plane and this is not an edge point. If we find only one straight line or no straight line, then this is an edge point.

Technique C. Surface Normal Approach:

Here we make use of the simple fact that the points on a plane have the same normal vectors and assume that the normals at each surface point are given. These are provided by our CAGD based 3-D modeling approach [7] and many times they are computed in a range segmentation technique [5]. Edge points on the surface of 3-D objects correspond to those points which lie on significantly different regions of the surface. Therefore, in this approach we calculate the change in normal vectors in the neighborhood of a point. If the normal vectors are changed significantly, then this signifies the presence of an edge point.

4. Results

In this section we present results on synthetic data, 3-D model data obtained from computer aided geometric design techniques and sensor data obtained from laser scanners. A comparison of the methods is also given.

Figure 1 shows two industrial objects, named "Green Piece" and the "Renault Piece" used in pattern recognition and computer vision research for 3-D object representation, recognition and manipulation. The 3-D range data for these objects was obtained by using both a laser scanner and computer-aided geometric design techniques [7]. The 3-D sensor data on Green Piece was obtained by using the White Scanner and the data on Renault Piece was obtained by using the INRIA laser scanner [5, 8]. These two objects had about 5000 and 2000 (x,y,z) surface points in one view.

The 3-D model data for these objects was obtained by using both a CAGD technique [6, 7] and the technique based on combining several views of the object [5]. CAGD technique uses the Alpha_1 experimental solid modeler system. It models the geometry of solid objects by representing their boundaries as discrete B-splines. It allows in a single system both high quality computer graphics and freeform surface representation and design. It uses a rational polynomial spline representation of arbitrary degree to represent the basic shapes of the models. Alpha_1 uses the Oslo algorithm [10] for computing discrete B-splines. Subdivision, effected by the Oslo algorithm, supports various capabilities including the computation associated with Boolean operations, such as the intersection of two arbitrary surfaces [18]. The details about the approach for building 3-D model from the sensor data are given in [4, 5].

Figure 2 shows the linked edge results on the objects shown in Figure 1 by using the gradient technique. White and gray edge points in this figure depict convex and concave edges respectively. In Figure 2(a) note that most of the holes, circles and surface scratches are correctly obtained, although a few of them show some gaps. In Figure 2(b) convex and concave edges are properly labeled.

Figure 3 shows the edge point results on the Green Piece by using the line fitting technique. These edge points can be thinned and linked like the images shown in Figure 2. The edge results are not as good as with the gradient technique. It is because of the higher sensitivity of this technique with respect to noise. Figure 4 shows the results on a synthetic cube. The cube data was rotated so that the points on the cube are not uniformally sampled. In this case very good edge results are obtained.

Figure 5(a) shows one view of the 3-D CAGD model of the Green Piece. All design aspects of the model can be seen very clearly on Evans & Sutherland PS300, 3-D graphics display. The model had about 12,000 points with a resolution of 0.1 inches. Figure 5(b) shows the edge points which can be further thinned and linked, etc. Very good results are obtained in this case. Figure 6(a) shows the points in one view of the 3-D CAGD model of Renault Piece. It had about 6000 points with a resolution of 0.2 inches. As a comparison to this design data in Figure 6(b) we show the 3-D model obtained by combing different views of the sensor data [5]. Figure 7 shows edge points detected in the top and front view of the Renault Piece data as shown in Figure 6(a). Note that a good set of edge points is obtained which can be thinned and linked and approximated by linear segments or arcs.

Figure 8 to 12 show results on cube model data (Figure 4(a)), Green Piece 3-D CAGD model data (Figure 5), and Renault Piece 3-D CAGD model data using the surface based normal technique. Note that good results are obtained. However, by comparing the results on the same data by line fitting technique, we find that line fitting technique produces better results.

Surface curvature of an object can be easily computed using normal vectors, (both for the CAGD model and the sensor data) and it is very useful in edge detection. An experiment evaluating the sensitivity of these techniques with respect to low signal-to-noise ratio is under investigation.

5. Conclusions

In this paper we presented three different approaches to detect edges in range images. The algorithms are not limited to single view of range image; they are applicable to the 3-D model surface data as well. The quality of results obtained gives us an indication that these results will be quite useful in developing matching techniques for 3-D shape recognition.

References

[1] Bajcsy, R.
Three Dimensional Scene Analysis.
Proc. 5th ICPR, Miami Beach, Florida :1064-1074, December, 1980.

[2] R.E. Barnhill and R.F. Riesenfeld (eds.).
Computer Aided Geometric Design.
Academic Press, New York, 1974.

[3] P.J. Besl and R.C. Jain.
Three-Dimensional Object Recognition.
Computing Surveys 17:75-145, March, 1985.

[4] B. Bhanu.
Shape Matching and Image Segmentation Using Stochastic Labeling.
Technical Report USCIPI 1030, USC Image Processing Institute, Los Angeles, California, August, 1981.

[5] B. Bhanu.
Representation and Shape Matching of 3-D Objects.
IEEE Trans. on Pattern Analysis and Machine Intelligence PAMI-6(3):340-351, May, 1984.

[6] B. Bhanu and T. Henderson.
CAGD Based 3-D Vision.
In *International Conference on Robotics and Automation.* IEEE, March, 1985.

[7] B. Bhanu, C.C. Ho and T. Henderson.
3-D Model Building for Computer Vision.
Technical Report UUCS85-112, Department of Computer Science, University of Utah, September, 1985.

[8] J.D. Boissonnat and F. Germain.
A new Approach to the Problem of Acquiring Randomly Oriented Workpieces out of a Bin.
In *Proc. International Joint Conf. on Artificial Intelligence*, pages 24-28. August, 1981.

[9] R.C. Bolles, P. Horaud and M. J. Hannah.
3DPO: A Three-Dimensional Part Orientation System.
In *Proc. 8th IJCAI*, pages 1116-1120. Karlsruhe, August, 1983.

[10] E. Cohen, T. Lyche and R.F. Riesenfeld.
Discrete B-splines and Subdivision Techniques in Computer-Aided Geometric Design and Computer Graphics.
Computer Graphics and Image Processing 14(2):87-111, October, 1980.

[11] Friedman, J.H., J.L. Bentley and R.A. Finkel.
An Algorithm for Finding Best Matches in Logarithmic Expected Time.
ACM Trans. on Math. Soft. 3(3):209-226, September, 1977.

[12] P. Horaud, R.C. Bolles.
3DPO's Strategy for Matching Three-Dimensional Objects in Range Data.
In *International Conference on Robotics*, pages 78-85. March, 1984.

[13] Inokuchi, W. and R. Nevatia.
Boundary Detection in Range Pictures.
In *Proc. 5th ICPR*, pages 1301-1303. Miami Beach, Florida, December, 1980.

[14] A. Mitiche and J.K. Aggarwal.
Detection of Edges Using Range Information.
IEEE Trans. on Pattern Analysis and Machine Intelligence PAMI-5:174-178, March, 1983.

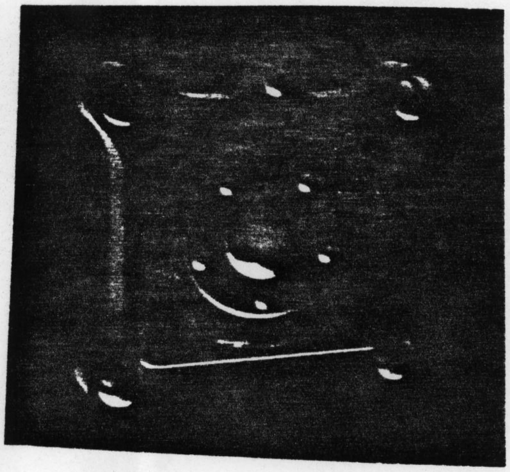
[15] R. Nevatia and K.R. Babu.
Linear Feature Extraction and Description.
Computer graphics and Image Processing 13:257-269, 1980.

[16] M. Potmesil.
Generating Models of Solid Objects by Matching 3-D Surface Segments.
In *Proc. 8th IJCAI*, pages 1089-1093. Karlsruhe, August, 1983.

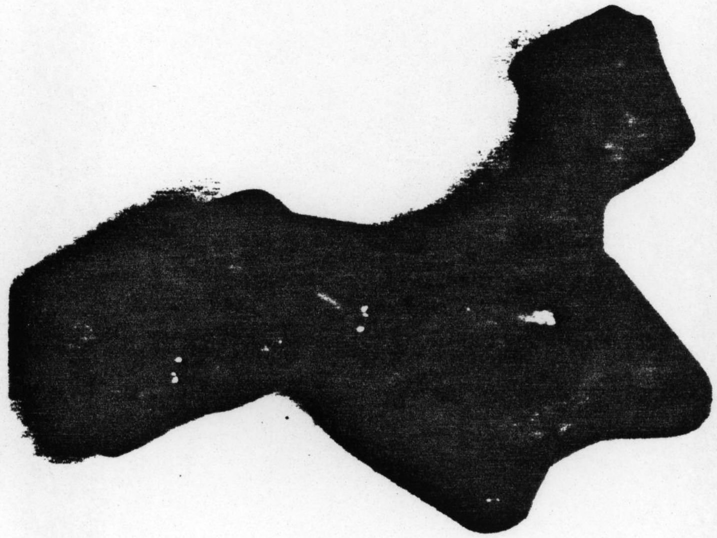
[17] K. Sugihara.
Range-Data Analysis Guided by a Junction Dictionary.
Artificial Intelligence 12:41-69, 1979.

[18] S.W. Thomas.
Modelling Volumes Bounded by B-Spline Surfaces.
PhD thesis, University of Utah, June, 1984.

[19] Zucker, S. and R. Hummel.
An Optimal Three-Dimensional Edge Operator.
In *Proceedings of the IEEE Conference on Pattern Recognition and Image Processing*, pages 162-168. August, 1980.

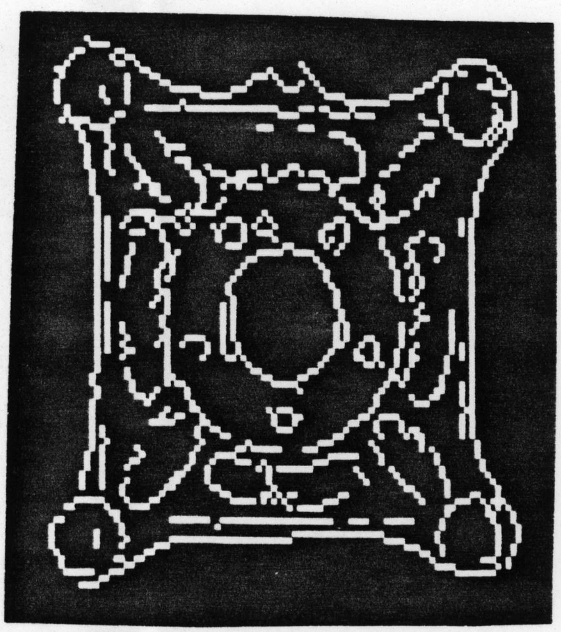


(a) Green Piece

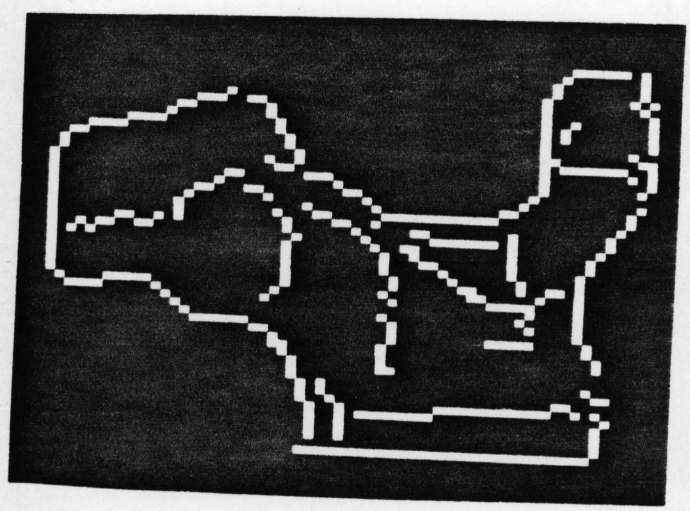


(b) Renault Piece

Fig. 1 Objects used for finding edges. They were designed using CAGD techniques and they were also sensed using a laser range finder.

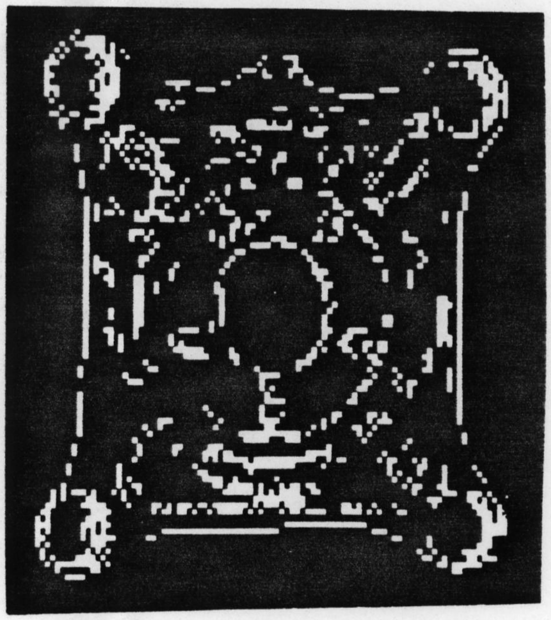


(a) Green Piece

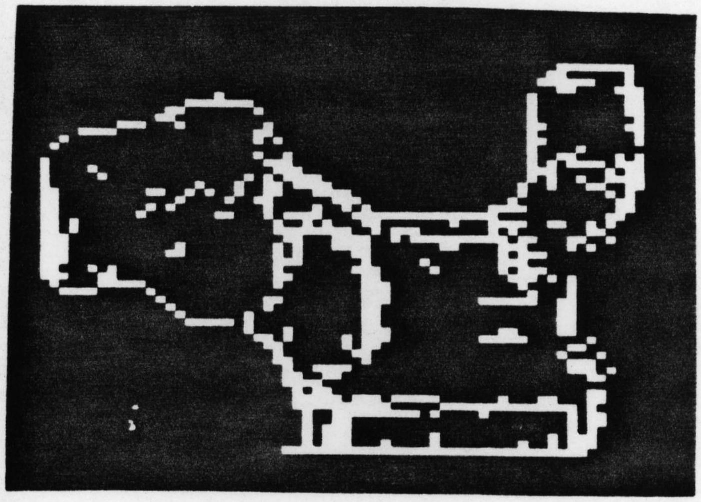


(b) Renault Piece

Fig. 2 Linked edges using the Gradient technique. White and gray edge points depict convex and concave edges respectively.

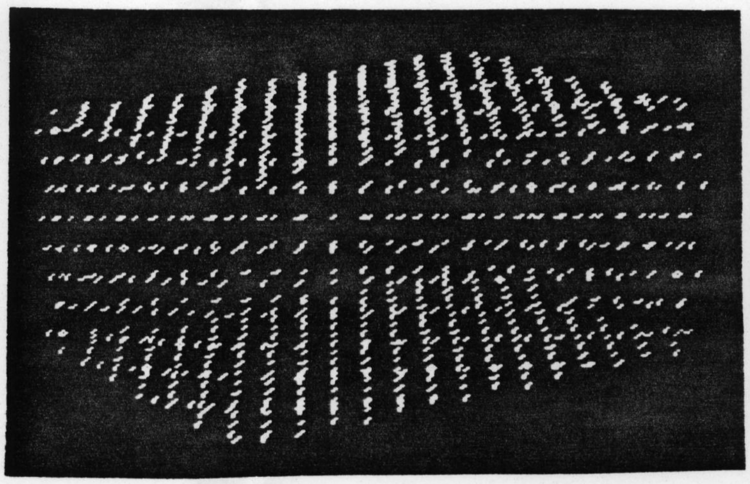


(a) Green Piece

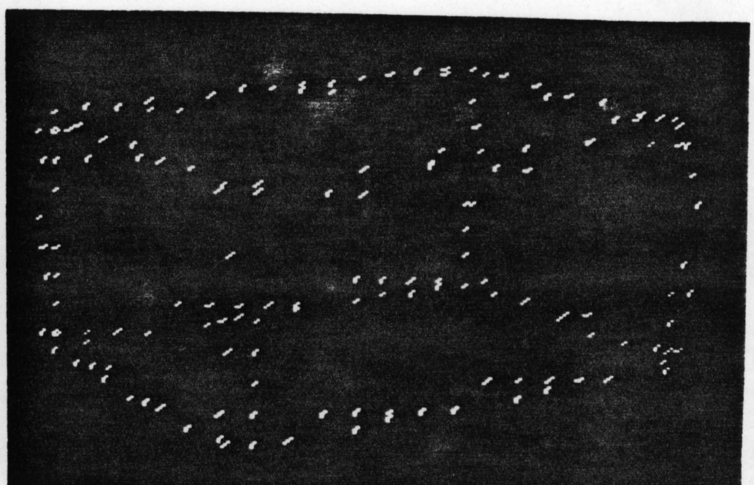


(b) Renault Piece

Fig. 3 Edge points obtained by using the line fitting technique.

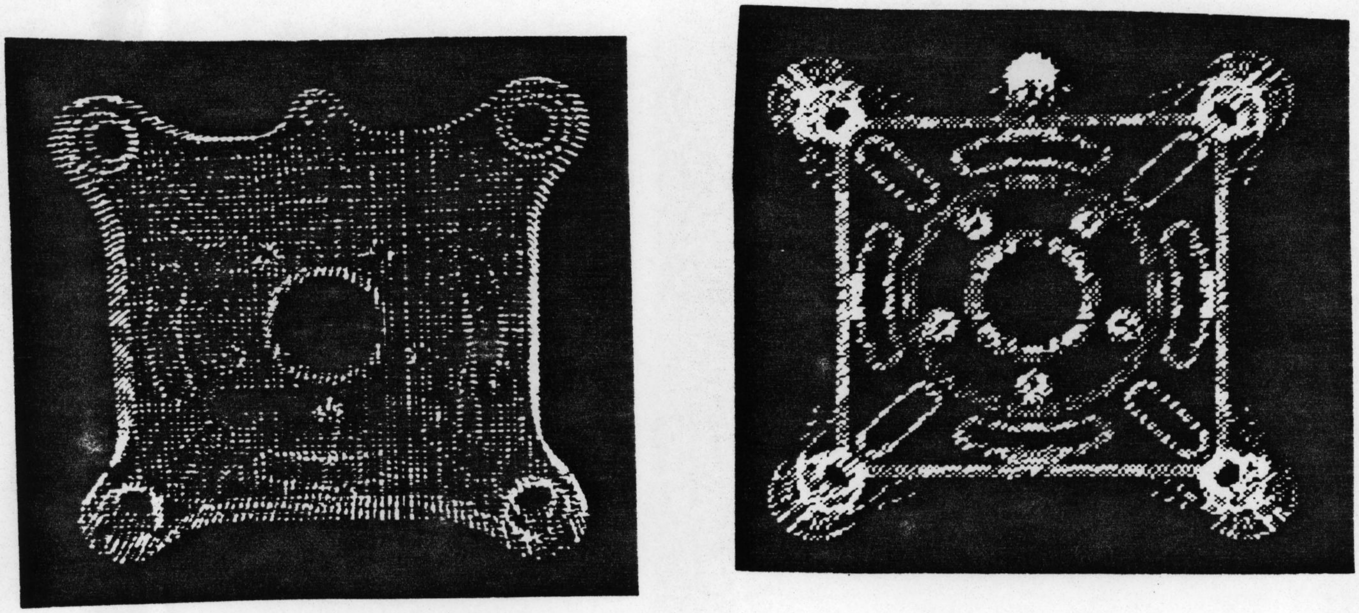


(a) Rotated 3-D cube data



(b) Edge points

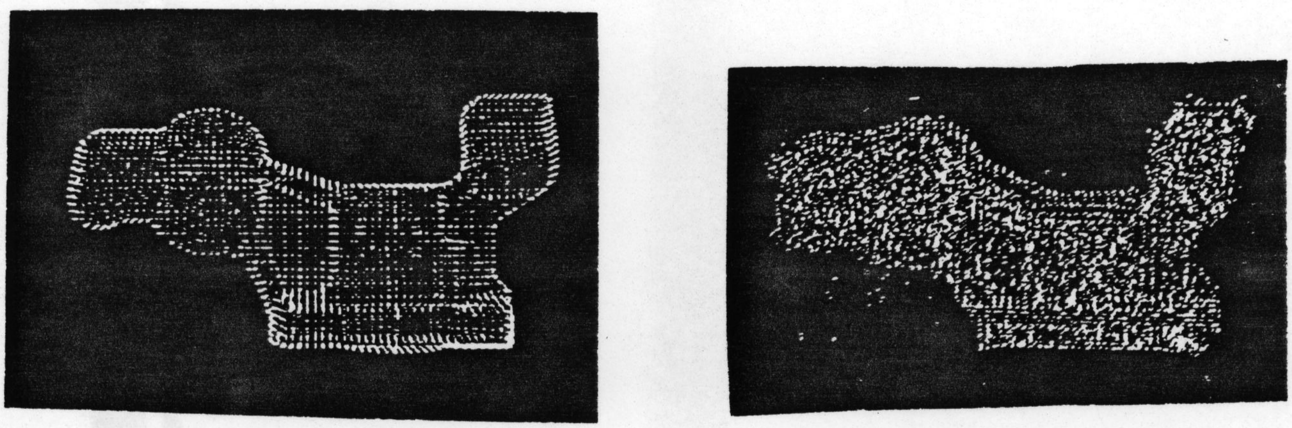
Fig. 4 Edge points obtained on the rotated synthetic cube data by using the line fitting technique.



(a) 3-D CAGD model data

(b) Edge points

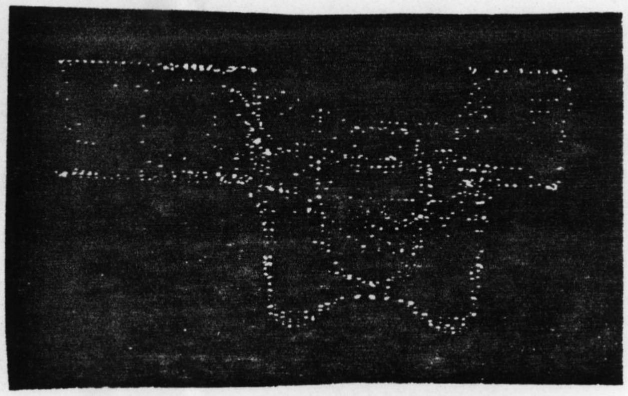
Fig. 5 Edge points obtained on the 3-D Green Piece CAGD model data by using the line fitting technique.



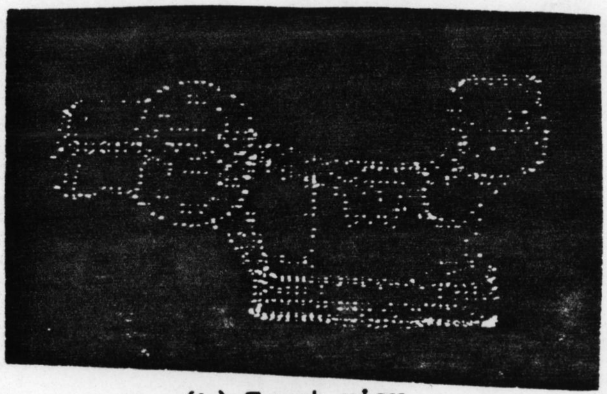
(a) 3-D CAGD model

(b) 3-D sensed model

Fig. 6 One View of 3-D model obtained by using CAGD techniques and the model built by combining several views of the sensed data.



(a) Top view



(b) Front view

Fig. 7 Two views showing the edge points obtained by using the line fitting technique.

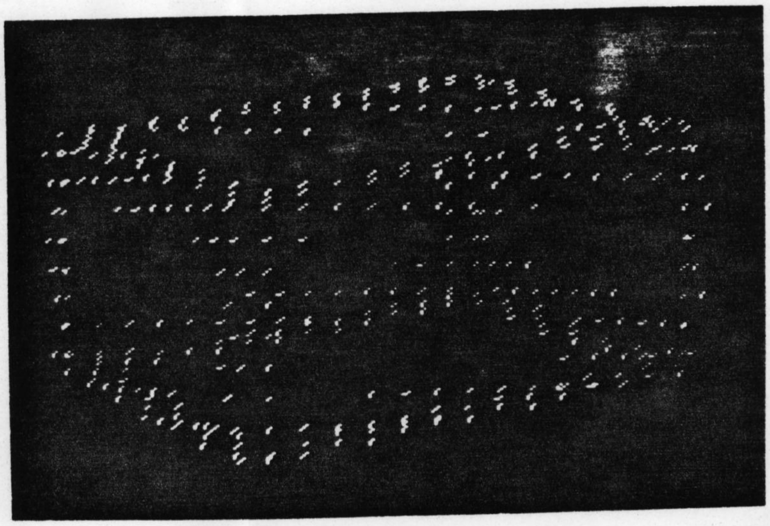


Fig. 8 Edge points obtained on the rotated synthetic cube data by using the surface normal based technique.

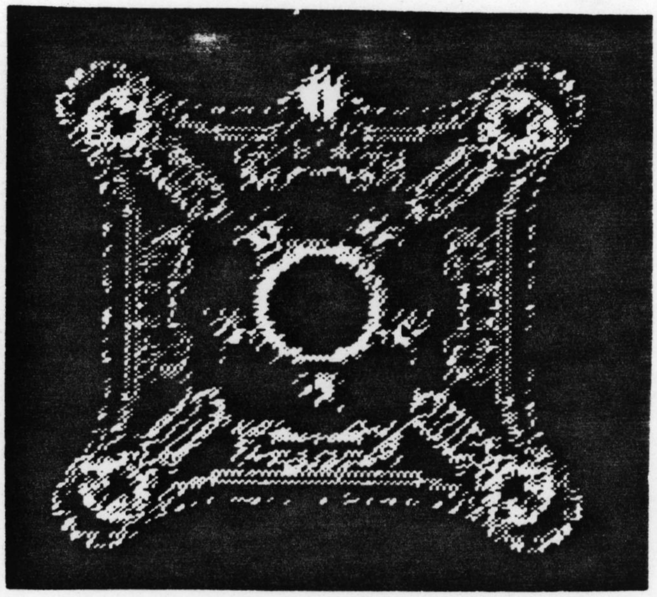
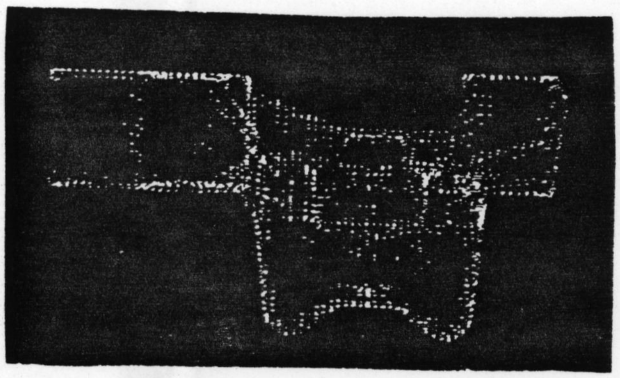
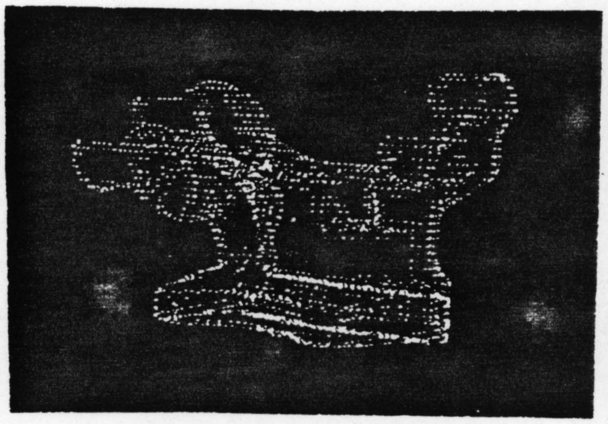


Fig. 9 Edge points obtained on the Green Piece CAGD model data by using the surface normal based technique.



(a) Top view



(b) Right view

Fig. 10 Two views showing the edge points obtained using the surface normal based technique.

List of figures