

THREE-POINT SEED METHOD FOR THE EXTRACTION OF PLANAR FACES FROM RANGE DATA

Thomas C. Henderson*
Bir Bhanu**

INRIA
Domaine de Voluceau - Rocquencourt
B.P. 105 - 78153 LeChesnay, FRANCE

Abstract

A method is given for representing a three-dimensional (3-D) object as a set of planar faces. The points representing the complete surface of the 3-D object are obtained by combining the object points from a sequence of range data images corresponding to various views of the object. The planar faces are then determined by sequentially choosing three very close non-collinear points (the 3-point seed) and investigating the set of points lying in the plane of these three points. Two simple tests, one for convexity and the other for narrowness, ensure that the set of points is an object face. The method is applied to an industrial object, and the results are discussed.

1. Introduction

Three-dimensional scene analysis consists of providing a model for 3-D objects and a method for matching unknown objects with the model. We are only concerned here with a method for obtaining the information necessary to form a surface model of a 3-D object. Our approach to the analysis of 3-D range data images is to first extract the relevant 3-D objects as sets of 3-D surface points and then to work directly on these sets without regard to the original images.

This approach frees one from a particular image and is in the vein of much work in computational geometry (Toussaint [1] gives an excellent review of this field). The goal is to produce a complete description of the surface of a 3-D object in terms of either the points on the surface or a segmentation of the surface into faces. Such a

representation should be complete, that is, should sample the entire surface of the object and allow for matching of individual views of the object taken from any vantage point. An object is thus defined as a finite number of selected points in 3-space. However, only the geometrical position of each point is known; no topological information is available. This lack of an ordering on the points is quite different from the case of one-dimensional curves and thus leads to a radically different segmentation strategy.

The proposed method is a sequential region growing algorithm and works on sets of 3-D points. Planar convex faces are determined by sequentially choosing three very close non-collinear points and investigating the set of points lying in the plane of these three points. Two simple tests, one for convexity and the other for narrowness, ensure that the set of points is an object face. This set of points is then approximated by polygons. The algorithm is not applied directly to range data images, but rather to a set of object points. In the example presented, these points were extracted from several range data images corresponding to various views of the object. Of course, the points from various views were transformed suitably so that all the surface points were given in terms of the same coordinate system.

2. 3-D Object Models

The major 3-D representations are: constructive solid geometry, boundary representations and sweep representations. In the first approach, a solid is built by combining primitive solids using set operations. Boundary representations involve surface point descriptions or surface approximation by polygonal faces. Finally, sweep representations (or generalized cylinders) define a 3-D object by means of a 3-D space curve, known as the axis, and cross-sections of arbitrary shapes and sizes along the axis. Bajcsy [2] presents a good discussion of the relative merits of surface representations (a set of 3-D points, a polygonal network, curved surface patches of various kinds, quadric patches) and volume representations (cellular space, convex polyhedra, geometric forms, ellipsoids, cylinders, spheres) on the basis of storage cost, complexity, operation on representation, conversion from one representation into another, etc.

* Now with the Department of Computer Science, University of Utah, Salt Lake City, Utah 84112

** Now with Ford Aerospace and Communication Corporation, Newport Beach, California 92660

Our approach to analyzing 3-D objects is to model them as polyhedra. Once the structure of the model has been determined, unknown objects can be matched to this structure. This requires, however, a description of the object in terms of vertices, edges and faces.

Methods for segmentation of range data into faces can be classified as "region" or "edge" based just as in the segmentation of intensity images. Many researchers adopt a method which is most suitable for the input device. For example, Duda et al [3] describe a sequential procedure for determining planar surfaces in a scene from registered range and intensity data. The vertical and horizontal surfaces are obtained directly from the range data by a histogram analysis. Slanted surfaces are assumed to have constant intensity and are obtained from the reflectance image. Finally, an iterative procedure refines the initial set of surfaces found. Milgram and Bjorklund [4] find planar surfaces in a range image (consisting mostly of planar regions) to determine the actual sensor position with respect to a given reference model. Planar surfaces are found by fitting a least squares plane in the small neighborhood of each pixel. The orientation of the plane, its altitude from the sensor and goodness of fit are recorded as local features of each pixel. Finally, pixels are examined for their association with their neighbors to form connected components. When the range image consists of complex objects having non-planar surfaces also, use of this approach does not seem suitable. Underwood and Coates [5] describe a system for inferring 3-D surface descriptions for planar convex objects from a sequence of views (reflectance images), but the faces are determined from edge information. Ishii and Nagata [6] obtained the contour of an object by controlling a laser spot. Shirai [7] and co-workers have used region and edge based methods to represent polyhedra and simple curved objects. Popplestone et al [8] dealt with polyhedra and cylinders. Inokuchi and Nevatia [9] presented a technique for obtaining surface edges, effectively a 3-D edge detector, while Zucker and Hummel [10] described an optimal 3-D surface edge operator which produces a smooth surface separating adjacent volumes. The drawback of these techniques is that the edge responses must be grouped, thinned and linked in order to produce a reasonable object description in terms of coherent region. On the other hand, once the line segments are found, the theory of 3-D line semantics can be directly applied. It is possible to extract planar surface from single view range data images by extending the iterative end point fit method (see Duda and Hart [11]) from 2 to 3 dimensions. This may work well since the range z can be considered as a function of two spatial coordinates x and y . All the above techniques apply to one range image at a time. Combining results from several views is a major problem.

3. Three-Point Seed Method

In this section we present an algorithm for surface approximation by polygons. Representing a 3-D object as a set of planar faces approximated by polygons is a two step process. In the first step

we find the set of points that belong to various faces of the object using the 3-point seed algorithm and in the second step approximate the face points obtained in step 1 by polygons.

The 3-point seed method for the extraction of planar faces from range data is a sequential region growing algorithm. It is not applied directly to range images, but to a set of points. It is not restricted to a single view range data image, but applicable to a composite object and does not require the ordering of points. It finds the convex faces of the object, but the information exists to merge convex parts of non-convex faces. The method is ultimately tied to the sampling distance between points on the object.

In a well-sampled object, any three points lying within the sampling distance of each other (called a 3-point seed) define a plane (called the seed plane) which:

1. Coincides with that of the object face containing the points, or
2. Cuts any object face containing any of the three points.

A seed plane satisfying (1) results in a plane from which a face should be extracted, while a seed plane satisfying (2) should be rejected. Two simple conditions that characterize category (2) are: convexity and narrowness.

Let us consider the intent of the convexity and narrowness tests. Figure 1 shows an object which has two faces (labeled 1 and 2) lying in the same plane. Thus, the set of points lying on this plane will be reduced by convexity to just those points in face 1. If the 3-point seed is in face 1, otherwise, face 2. Note that face 3 (a non-convex face) will be segmented differently depending on the order in which the points in that face are examined. The second case in which convexity plays a role is shown in Figure 2. In this example (Figure 2(a)), the 3-point seed (the three circled *'s) straddles an edge, and consequently the points in the seed plane (shown by *) lie on a section through the object and not on any actual object face. The convexity condition reduces this set of points to the points shown in Figure 2(b). The peculiar property of such a reduced set s that its points all lie essentially on the line passing through the two most distant points in the set. This fact gives rise to the narrowness test, and any set of points which is "narrow" in this sense is rejected as a possible face. A semi-PASCAL version of this procedure is given as Algorithm 1.

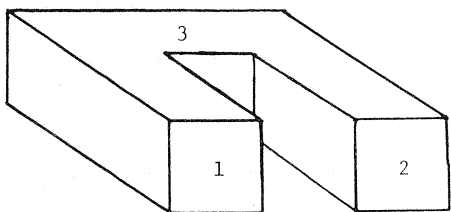


Figure 1: Face 1 and 2 lie in the same plane but are distinct; Face 3 is non-convex.

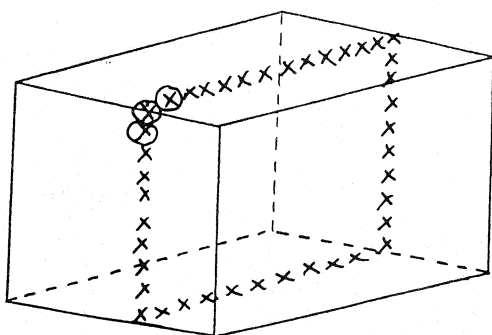


Figure 2a: The 3-point seed (the circled x's) straddles any edge and produces a plane which cross-sections the object. The convexity condition reduces the set of points to those shown in Figure 2b.

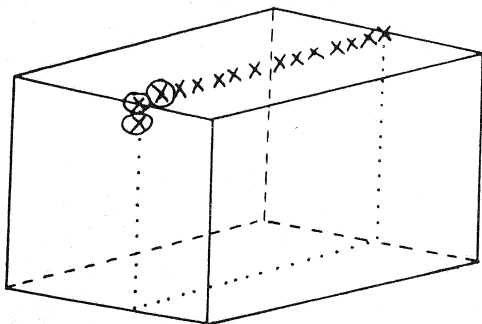


Figure 2b: The narrowness test excludes faces like this whose points all lie very near the same line.

```

procedure Three_point_seed;
begin
POINTS := {Xi}, i = 1,n;
for i := 1 to n-2 do
  if not Marked(Xi)
  then for j := i+1 to n-1 do
    if not Marked(Xj)
    then for k := j+1 to n do
      if not Marked(Xk) and
        Close_enough(Xi,Xj,Xk)
      then begin
        Set_plane(Xi,Xj,Xk,a,b,c,d);
        PLANE := Plane_pts(POINTS,a,b,c,d);
        FACE := Convex(PLANE,Xi,Xj,Xk);
        if not Narrow(FACE)
        then begin
          Store(FACE);
          Mark(FACE,POINTS);
        end;
      end;
    end;
  end;
end;

```

Algorithm 1. 3-Point Seed Algorithm.

POINTS is the set of n object points described by their (x,y,z) coordinates; i.e., $X_i = (x_i, y_i, z_i)$. The predicate $\text{Marked}(X_i)$ returns true if X_i has already been associated with a face and false otherwise. Once a point is associated with a face, it cannot be part of a 3-point seed or new face. The predicate Close-enough returns true if the 3 unmarked points are near enough to each other.

Once the 3-point seed has been found, the seed plane, PL, passing through these points can be analyzed for the existence of an acceptable face containing the seed points. The coefficients (a,b,c,d) characterizing the seed plane, PL, are used to determine the set PLANE which consists of all unmarked points in POINTS "lying" within a certain distance of the plane PL. The set PLANE is then reduced to the maximal convex set in PLANE containing the seed points; this convex set is called FACE (see Rosenfeld [12] for digital convexity). If the points in FACE constitute a face which is not too narrow, then the information concerning the face is stored, and the points in POINTS belonging to FACE are marked. The predicate Narrow returns true if all the points in the face lie very close to the line (in 3-D space) which passes through the two most distant points of FACE. Such a set of points occurs when the seed straddles an edge or vertex of the object. The points in the previously found faces are checked for inclusion in the current face. This determines the points that belong to more than one face and provides the knowledge about the neighbors of a face and relations between faces. Although some edge points and vertices will be known, an independent step is required to obtain polygonal approximations to these faces.

The worst-case complexity of the 3-point seed algorithm is $O(n^4)$. There are $O(n)$ 3-point seeds to consider since each object point can be grouped in twelve ways with its neighbors to produce a seed. For each 3-point seed considered, the largest cost is in the convexity test which is of

$O(n^3)$ since every time a point is added to the FACE list, all rejected points in that plane have to be reconsidered. In practice, however, the complexity is more dependent on the number of faces than the number of points. Moreover, the convexity test can be organized around a k-d tree, and the complexity of this test reduced to $O(n \log n)$. Only convex faces are found for the object; however, the information exists to merge the convex parts of non-convex faces. Another problem is that the seed points are chosen sequentially, and this influences the selection of faces. However, if the interior of every face is represented by enough sample points, the major faces will always be found. Finally, the narrowness condition rejects small or narrow faces, i.e., a face one or two points wide may be rejected. This can always be overcome by increasing the sampling rate.

The 3-point seed algorithm can be viewed as a generalization of a method for segmenting one-dimensional curves (in any dimension). Choose two close points, x_0 and y_0 , on the curve. Let $S = \{x : x \text{ lies on the line } (x_0, y_0) \text{ and } x \text{ not in any segment yet found}\}$. Reduce S to the maximal convex set containing x_0 and y_0 . Mark all points in the resulting segment S , and start all over. This algorithm does not take advantage of the ordering of the points on the curve; as do most piecewise linear schemes. Thus, this method easily generalizes to 2-D surfaces, and is implemented here as the 3-point seed algorithm.

The 3-point seed method can be viewed as a special case of the Random Sample Consensus paradigm (RANSAC) of Fischler and Bolles [14]. RANSAC is a model fitting method which uses as small an initial data set as feasible and enlarges this set with consistent data when possible. They applied the technique to the location determination problem with great success.

4. Data Acquisition

The object surface point data used in this study was obtained with a laser ranging system whose principle is shown in Figure 3. A laser emits a beam of ruby red light which is reflected by a mirror which rotates and sweeps the beam along the x-axis to produce one scan line. The beam is reflected from the object, and the z-distance is calculated from the location of the maximum response in each bank of detectors (and the orientation of the mirror if necessary). The platform on which the object rests can be raised or lowered (this is the y-axis) and can also be rotated (around the y-axis). The sampling distances used here are 3.0 mm in the x-axis, 2.0 mm in the y-axis, and an accuracy of 0.5 mm in the z-distance is achieved.

This data is in the observer centered coordinate system. To create a 3-D model of the object, an object centered representation is required. This is computed by marking the zero position for the x-axis and y-axis and obtaining a reference value for the z-axis on the platform on which the object rests. Thus, all the points can be transformed into the same coordinate system. For the object

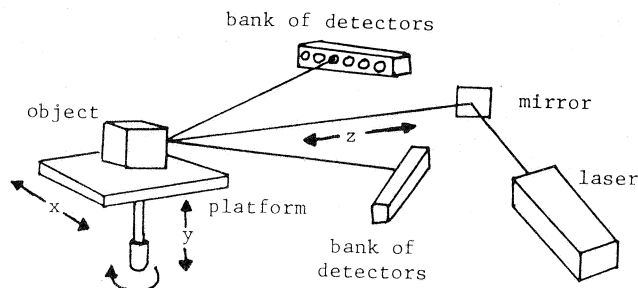


Figure 3: Laser Ranging System

studied, a range data image was produced for every 30 degrees rotation of the object around the y-axis. Finally, top and bottom views of the object were taken. These last two views were put in correspondence with the other views by using several control points on the object to compute the transformation.

5. Experiments

Both synthesized objects and an industrial piece used in an automobile suspension system have been analyzed with the proposed method for surface approximation by polygons. Results obtained on synthetic cubes were exactly the faces of the cubes, and faces found on synthetic spheres were reasonable. A more difficult case is that of the complicated casting shown in Figure 4. This object does not contain any major horizontal or vertical surfaces. Figure 5 shows the 14 range views as gray scale images. In this figure the lighter a point is the farther away it is. After thresholding the background points, each view shown in Figure 5 had about 2000 points. Surface points for the composite object were obtained by rotating the points in the 12 views Figure 5(a) to 5(l), and control points were used to transform the top and bottom views. For each view, the required transformation was applied, and the distance between the transformed point and the points already in the list (in the beginning just the 0 degree view points) was used to add previously unrepresented points to the list. With all the views combined, 8334 points were produced for the composite object.

The 3-point seed method was applied to the 14 individual views shown in Figure 5 and to the composite object. Figure 6 shows the faces found for the 0 degree view (Figure 5(a)) of the object. The points that could not make up a face of at least 20 points were rejected. The area of rejected points falls either on jump points resulting from large z-distance change with correspondingly little x or y change, or they occur in extremely uneven parts of the surface of the object. A rejected point lies inside some of the faces because it has been missed in the process of data acquisition. As can be seen, most of the faces found are reasonable. The object has major

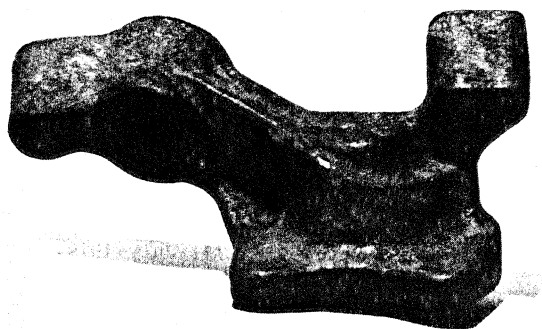


Figure 4: Automobile Piece Analyzed

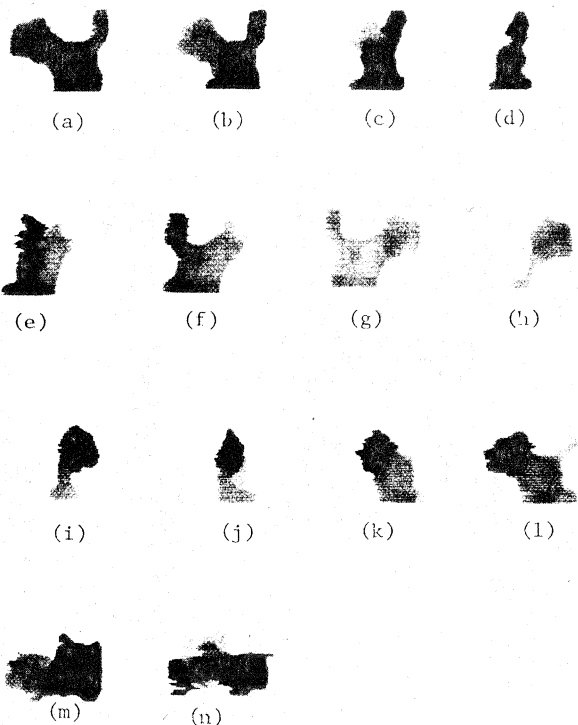


Figure 5: Range Data Images

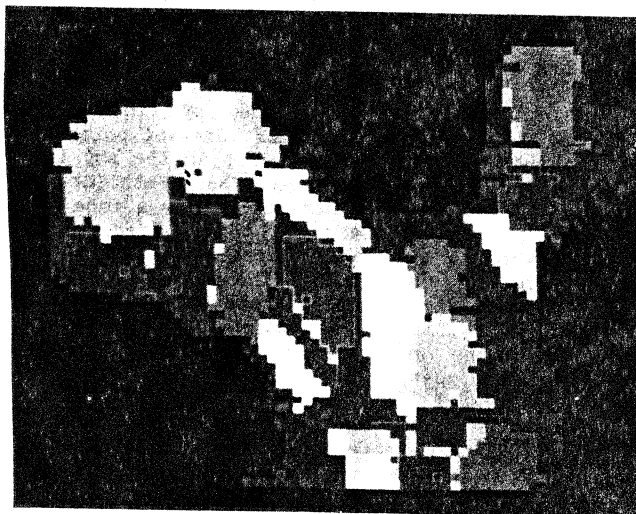


Figure 6: Faces Found for Figure 5(a)

curved surfaces that were split symmetrically into different faces. Edge points are not very well detected.

In the beginning we find faces which have at least some maximum number of points and iteratively decrease the necessary number by some small amount until we reach a minimum number of points allowed. The faces having less than this number are rejected. A total of 85 faces were found for the composite object, and their distribution fits well with the results from the individual views.

6. Discussion

The 3-point seed algorithm alludes obliquely to several thresholds which are directly related to the sampling distances. First, three points are close enough together to be chosen as seed points if each pair of the three is within $\sqrt{dx^2 + dy^2}$, where dx and dy are the distances between samples in the x and y directions, respectively. Call this distance ϵ_{ps_1} . The second threshold, ϵ_{ps_2} , is the distance a point can be from a plane and still be considered in the plane. If ϵ_{ps_2} is too low, neighboring faces will most likely have no common points, making the relations between the faces harder to find. The third threshold, ϵ_{ps_3} , is that used to determine face convexity. The check for convexity involves finding the midpoint of the candidate point and each point already in the face (initially just the seed points). If for some point of the face, the midpoint is not close enough to any point already in the face, then the candidate point is rejected. Of course every time a new face point is found, all rejected points must be reconsidered. The threshold ϵ_{ps_3} cannot be

chosen smaller than $\sqrt{dx^2+dy^2}/2$ or else no set of points is convex.

Finally, narrowness is decided by finding the line passing through the two most distant points in the set and checking if some point in the set lies far enough away from this line. The threshold ϵ_{ps_4} for this distance determines how narrow a face can be. For the automobile piece, $dx=3$, $dy=2$, $dz=3$, and the thresholds chosen were: $\epsilon_{ps_1} = \sqrt{19}$, $\epsilon_{ps_2} = 3$, and $\epsilon_{ps_3} = \sqrt{9.5}$. The choice of $dz = 3$ is due to the maximum sampling rate that the z-axis can undergo. The threshold ϵ_{ps_4} , however, is dependent on the orientation of the line in space between the two most distant points; in particular, $\epsilon_{ps_4} = dx \cdot \cos(a) + dy \cdot \cos(b) + dz \cdot \cos(c)$, where $\cos(a)$, $\cos(b)$ and $\cos(c)$ are the direction cosines of the line.

The method is applied in stages; the largest faces (in terms of the number of points in the face) are found first, then smaller faces on down to some minimum size. This is necessary in order to limit the fragmentation of large faces near their extremes and also allows one to avoid refining the planes found. Unfortunately, the unevenness of the surface of the object can still lead to poor face selection; however, the peculiarities of the object to be modeled can be accounted for by the proper choice of thresholds, and the tradeoff involved between the number of faces and the quality of representation can be balanced. It may be useful to restrict the set of 3-point seeds. For example, a "plane detector", i.e., a filter with high response where the data fits a plane well, can be passed over a range data image to produce the most likely locations for seed points. More evidence for the "goodness" of the faces found is the successful application of this method as the basis for a 3-D segmentation and matching system described by Bhanu [14].

7. Summary

A new technique for the extraction of planar faces from a set of 3-D points is proposed. This method is of worst-case time complexity $O(n^4)$ and sequential in nature. The underlying physical motivation for the method is given. The application of the method to synthetic and real world 3-D objects is described. The major advantage of the method is that it is applicable to a composite object, and not restricted to single view range data images. Information on the neighboring faces is also easily obtained. Once the planar faces are found, they are approximated by polygons.

References

- [1] G.T. Toussaint, "Pattern Recognition and Geometrical Complexity," Proc. 5th ICPR, Miami Beach, Florida, pp. 1324 - 1347, Dec. 1980.
- [2] R. Bajcsy, R. "Three Dimensional Scene Analysis," Proc. 5th IJCAI, Miami Beach, Florida, pp. 1064 - 1074, Dec. 1980.
- [3] R.O. Duda, D. Nitzan and P. Barrett, "Use of Range and Reflectance Data to Find Planar Surface Regions," IEEE Trans. on Patt. Anal. Mach. Intell. Vol. PAMI-1, No. 3, pp. 259 - 271, July 1979.
- [4] D.L. Milgram and C.M. Bjorklund "Range Image processing: Planar Surface Extraction," Proc. 5th Int. Conf. on Pattern Recognition, Miami Beach, Florida, pp. 912 - 919, Dec. 1980.
- [5] S.A. Underwood and C.L. Coates, Jr., "Visual Learning from Multiple Views," IEEE Trans. on Computers, Vol. C-24, No. 6, pp. 651 - 661, June 1975.
- [6] M. Ishii and T. Nagata, "Feature Extraction of Three-Dimensional Objects and Visual Processing in a Hand-Eye System Using a Laser Tracker," Patt. Recog., Vol. 8, pp. 229 - 237, 1976.
- [7] Y. Shirai and M. Suwa, "Recognition of Polyhedra with a Range Finder," Proc. 2nd IJCAI, pp. 80 - 87, 1971.
- [8] R.J. Popplestone, C.M. Brown, A.P. Ambler, G.F. Crawford, "Forming Models of Plane-and-Cylinder Faced Bodies," Proc. 4th IJCAI, pp. 664 - 668, 1975.
- [9] S. Inokuchi and R. Nevatia, "Boundary Detection in Range Pictures," Proc. 5th ICPR, Miami Beach, Florida, pp. 1301 - 1303, Dec. 1980.
- [10] S.W. Zucker and R.A. Hummel, "An Optimal Three-Dimensional Edge Operator," Proc. IEEE Conf. on Patt. Recog. and Image Proc., pp. 162 - 168, 1980.
- [11] R.O. Duda and P.E. Hart, Pattern Classification and Scene Analysis, New York, Wiley-Interscience, 1973.
- [12] A. Rosenfeld and A. Kak, Digital Picture Processing, Academic Press, New York, 1976.
- [13] M.A. Fischler and R.C. Bolles, "Random Consensus: A Paradigm for Model Fitting with Applications in Image Analysis and Automated Cartography," CACM, Vol. 24, No. 6, pp. 381 - 395, June 1981.
- [14] B. Bhanu, "Shape Matching and Image Segmentation Using Stochastic Labeling," USC-IPI Tech. Rep. 1020, Sept. 1981.