CAD Based 3-D Models for Computer Vision

Bir Bhanu, Chih-Cheng Ho and Sungkee Lee

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

Abstract: Model based recognition is one of the key paradigms in computer vision and pattern recognition. However, at present there is an absence of a *systematic approach* for building geometrical and functional models for a large class of 3-D objects used in industrial environments. In this paper we present a Computer-Aided Design (CAD) based approach for building 3-D models which can be used for the recognition and manipulation of 3-D objects for computer vision applications. We present the details of the design on a relatively simple object named "Green Piece," and the complex automobile part, "Renault Piece," used in computer vision and pattern recognition research. We also present examples on the derivation of four different representations (surface points, surface curvatures, edges, arcs and local features, and volumes and sweeps) from the CAD designs so as to build computer vision systems capable of handling multi-class objects and employing *multiple representations.*

1. Introduction

The emergence of Computer Integrated Manufacturing (CIM) technology has provided opportunities and challenges to use geometrical and functional models of real-world 3-D objects for the task of visual recognition and manipulation of these objects by robots [3]. CIM technology provides the database of objects as a byproduct of the design process. It allows the model-based recognition of 3-D objects to be simulated even before these objects are physically created. In this paper we present our ongoing work in defining how these designs could be modified in novel ways so that they can be made useful for the task of recognition and manipulation.

A formal CAD system contains an interactive user interface, graphic display utilities, model analysis tools and automatic manufacturing interfaces. It is a suitable environment for design purpose. However, the models it generates do not contain features which are important in computer vision. A *systematic approach* to build vision models employing multiple representations [3] is to construct them from the CAD database and incorporate those features which are crucial for object recognition and manipulation. Most of the work in 3-D objects recognition uses only one representation and a matching technique based on this representation. CAD based approach allows the construction of models employing several representations; thus it is able to handle a wider class of objects. Moreover, it allows different parts of the *same* object to have different representations. These multiple representations and multiple matching techniques based on these representations are very useful in an automated working environment where a robot equipped with multi-sensors is to be operated. Four basic representations derived from CAD designs are:

1. Representation of objects by *surface points.* These points can be sampled at a desired resolution and the information regarding the neighborhood and surface normal is provided which can be used in building higher level description of the object in terms of planar, spherical, cylindrical and conic surface patches.

2. Representation of objects by *surface curvatures.* The local surface can be characterized by curvatures alone. In B-splines based Alpha_1 CAD system [8], derivatives are embedded inherently in its control mesh and knot vectors. Surface normals and curvatures can be obtained as a byproduct of many B-splines computations. These are intrinsic surface characteristics.

3. Representation of objects by surface *edges, arcs* and *local features.* Edges and arcs can be extracted as boundaries of surface patches used in the design process. A more precise way is to define edges as the local extrema of curvature or to use 3-D edge detection techniques.

4. Representation of objects by *volumes* and *sweeps.* From the CAD design, we can construct a Constructive Solid Geometry (CSG) like representation in which a functional information can be incorporated. Also, the CAD model can be converted into a generalized cylinder model by applying an object decomposition technique followed by an axis and cross-section extraction procedure.

In this paper we present these approaches for model building using the Alpha_1 CAD system developed at the University of Utah.

2. Model Building Using the Alpha_1 CAD System

Alpha_1 models the geometry of solid objects by representing their boundaries as non-uniform rational B-splines. It uses the Oslo algorithm [8] for knots insertion. Rational B-spline is an ideal design tool: it is simple yet powerful and all quadratic surfaces which are used as primitives in CSG can be represented exactly. Other advantages include good computational and representational properties of the spline approximation: the variation diminishing property, the convex hull property and the local interpolation property. Alpha_1 supports several modeling paradigms, including direct manipulation of the B-spline surfaces, creation and combination of primitive shapes, and high-level shape operators such as bend, twist, warp and sweep. It also allows set operations on surface patches which make the modeling task easy and complete. Here are some guidelines in using Alpha_1: (1) Analyze the object, a complex object is decomposed into simpler parts which are designed more easily. (2) Make a precise measurement of parameters. (3) Maintain validity of the model by setting correct orientation and adjacency information of each patch. (4) Perform the appropriate transformations and Boolean operations.

To design simple objects such as the "Green Piece" (Fig. 1) which has many local features, we build the complete object in a stepwise manner. First, we design the plate and all holes as in Fig. 2(a), then the dent part and scratches of Figs. 2(b) and 2(c). To design these parts, we first design curves using B-splines and then use various high level operators for surface construction, such as revolving a curve about an axis, extruding a curve in some direction and filling the surface between two curves. There are seven threads in the Green Piece. Each of these is designed by filling two surfaces between two twisted curves. Fig. 2(c) shows the center one, the others are similar except their radii and pitches. Fig. 3 is the completed CAD model. This design can be used in manufacturing the green piece on a numerically controlled milling machine.

For object like the Renault Piece (Fig. 4) which contains sculptured

---

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

freeform surfaces, it is divided into a set of simpler parts. Here we divide it into five sub-parts: (a) neck, (b) right head, (c) back bump, (d) left head, and (e) base plate (Fig. 5). For the right head, left head and back bump, we find all sharp edges and then construct the surfaces from them as before. For the base plate and the neck part, we need some pseudo sharp edges which are the intersection of the surface planes. Then we construct these surfaces but leave small gaps between them where cubic B-spline patches are used to produce the rounded edges. Fig. 6 shows the intersection curves of these parts, which are computed to obtain the complete object using set operations performed by the combiner in Alpha_1. Fig. 7 is the completed CAD model for the Renault Piece.

3. Representation of Objects by Surface Points

The set operations over B-spline surfaces are not closed, parts of the CAD model are represented as polygons. Our strategy is to subdivide all the B-spline surfaces into polygons so as to make the problem uniform. By applying a contour filling algorithm, we get interior line segments of the polygons, and extract points along these segments at a desired resolution. The resolution is defined as the maximum distance between any two adjacent points.

An edge based contour filling algorithm is described in [15] which requires expensive preprocessing of contours. It is used in applications, such as surface shading, where the same contour is used repeatedly. In our case, the number of polygons in a model is very large and we extract only a small number of points from each one of them. Instead of sorting and marking the edges, we apply a new algorithm [4] which uses topological information of the intersection points and characterizes them as start, middle or end points of segments. Then the segments between all start/end point pairs are inside the polygon. Since the number of vertices is usually much larger then the number of intersection points for one line segment, its time complexity is linear to the number of vertices in the polygon in the average case.

The above 2-D algorithm with some modifications is applied to 3-D polygons, The given resolution is ensured by finding the maximum projection plane of each polygon among either x-y, y-z or z-x planes from its normal vector and using cutting planes orthogonal to it. Thus the distance between any two adjacent points is less than the grid spacing times $\sqrt{3}$. Fig. 8 shows the surface points extracted from the models of Green Piece and Renault Piece at 0.2 inch resolution. In Fig. 9, we show the surface normals of Green Piece model at 0.4 inch resolution, which are computed using bi-linear interpolation of normals at vertices. Fig. 10 exhibits two samplings of surface points of Renault Piece and its real range data. These synthetic data are taken from only one view and can also be used to simulate the real range data.

4. Representation of Objects by Surface Curvatures

The use of surface curvatures as intrinsic characteristics of surface shape is adopted from differential geometry to computer vision recently. Major applications are: (1) Use the sign of Gaussian and mean curvatures to construct the topographic primal sketch [11] and to characterize the type of surfaces [2]. (2) Use lines of curvature to determine planar patches and umbilical regions [7]. (3) Use the zero crossing of Gaussian curvature to find roof edges and the extrema of principal curvatures to find step edges [1]. Another application is to construct the Extended Gaussian Image (EGI), which is defined as the mapping associated with the inverse of the Gaussian curvature at a surface point with the corresponding point on the Gaussian sphere [12].

To compute the four basic types of surface curvatures: Gaussian, mean, maximum, and minimum, we need the first and second derivatives of the surface and the unit surface normal vectors [10]. In Alpha_1, the basic surface type is tensor product B-spline patch. It can be written as [9]:

$$s(u,v)=\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{d}_{i,j}B_{i,k}(u)B_{j,l}(v)$$

which is completely specified by:

k and l:	Orders of the spline. These are the greatest degree of any of the polynomial pieces plus one in u and v directions respectively.
$\mathbf{d}_{i,j}$:	Control mesh, is a matrix of spatial points.
$B_{i,k}$ and $B_{j,l}$:	Basis functions. They are the weight of each of the control points when they are blended to get the surface points. Each of them is defined by an ordered set of non-decreasing scalar values, the knot vector.

A convenient way to think of the tensor product surface is to think of the rows or columns of the control mesh as a set of individual B-spline "control curves" (with one knot vector and the order associated with each of them). The other knot vector and its order then describe how these curves will be blended to form the surface. The derivatives of this B-spline surface is another tensor product B-spline surface with a lower order formed by the differentiation of its "control curves". For example the first partial derivative of surface s(u,v) w.r.t. to variable u is:

$$\frac{\partial s(u,v)}{\partial u}=\sum_{i=1}^{n+1}\sum_{j=1}^{m}(k-1)\frac{\mathbf{d}_{i,j}-\mathbf{d}_{i-1,j}}{u_{i+k-1}-u_i}B_{i,k-1}(u)B_{j,l}(v)$$

$$\text{where } \mathbf{d}_{0,j}=\mathbf{d}_{n+1,j}=0$$

Higher order derivatives are found by successive differentiations. The unit normal vector is found from the cross product of the first partial derivatives (tangent vectors). Since it requires rational computations in normalization, there is no closed form of the curvatures in B-spline. In Fig. 11, we show the result of the computation of the four basic types of curvatures of a Coons patch (Dark area means negative value and bright area means positive value). Symbolic information is then derived from these curvatures. As an example, Fig. 12 shows the edge points found by simply requiring the absolute value of both principal curvatures to be above a threshold.

5. Representation of Objects by Edges, Arcs & Local Features

Three techniques are used to detect edges in range data.

Technique 1 - Gradient Approach: A second derivative formulation for discrete case is used to calculate the magnitude and directions of an edge at each point similar to the work by Sugihara [17]. We used four 3x3 edge operators to calculate them. In general, most edges in intensity images are step edges. But the edges in range images can be either step or roof edges. As a result we cannot get all the edges in a range image by edge operators used on intensity images. Preferably, for range images an edge operator should be sensitive to roof edges.

The four operators, each of different direction (0°, 45°, 90°, and 135°), are given below.

$$\begin{matrix}1&1&1\\-2&-2&-2\\1&1&1\end{matrix}\qquad\begin{matrix}1&1&-2\\1&-2&1\\-2&1&1\end{matrix}\qquad\begin{matrix}1&-2&1\\1&-2&1\\1&-2&1\end{matrix}\qquad\begin{matrix}-2&1&1\\1&-2&1\\1&1&-2\end{matrix}$$

The performance evaluation of these operators with respect to signal-to-noise ratio is given in [5].

Technique 2 - 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. After getting neighboring points in 3-D space, we calculate the unit direction vectors from the center point to its neighboring points. The two of these direction vectors lie on a straight line if they point to exactly opposite directions. If we find two or more straight lines within a certain threshold (related to the differences of the direction vectors), then the center point and all its neighboring points lie on a plane and it is not an edge point. Conversely, if we find only one or no straight line , then it is an edge point.

Technique 3 - Surface Normal Approach: Here we make use of the simple fact that the points on a plane have the same normal vector and assume that the normal at each of the surface points is given. These are provided by our CAD based 3-D modeling approach [4] and many times they are computed in a range segmentation technique. In this approach we calculate the change of direction of normal vectors in the neighborhood of a point. Here we have used 8 neighboring points. If the difference of unit normal vectors is greater than a certain threshold value, then it signifies the presence of an edge point.

Fig. 13 shows range data on the "Green Piece" and the "Renault Piece". The 3-D sensed 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. These two objects have about 5000 and 2000 surface points in one view, respectively. Fig. 14 shows the linked edge results on the objects shown in Fig. 13 by using the gradient approach. White and gray edge points in this figure depict convex and concave edges. In Fig. 14(a), note that most of the holes, circles and surface scratches are correctly obtained, although a few of them show some gaps. In Fig. 14(b) convex and concave edges are properly labeled. Fig. 15(a) shows the results on the data of Fig. 8(a) using the line fitting technique. Very good results are obtained in this case. Fig. 15(b) shows the results using the surface normal based approach.

The feature values of the line segments, circles and ellipses are calculated by using the Hough transform technique. First, the straight line equations are calculated. Then feature values of the line segments are obtained. These include starting point, ending point and length of the segment. The minimum distance and the angle between line segments are also calculated. To find the circles, we used parametric equations, $x = r^*\cos(\theta)$ and $y = r^*\sin(\theta)$, of a circle. The edge direction is used in transforming each edge point into parametric space in order to reduce computing time and to get better results. Here, the increment of θ is proportional to $1/r$ and thus the threshold value is proportional to the radius r to take local maximum of the array of accumulators. The arcs are found in a similar fashion and their properties are computed. Finally, the distance between the center points of circles are calculated. A general ellipse equation needs five independent parameters, and it is impractical to apply Hough transform to get the equation of an ellipse because of the memory space and the computing time. In this step, all the edge points which correspond to line segments, circles and arcs are removed from the edge image to reduce the computing time and to get better results. The modified Hough transform technique by Tsuji and Matsumoto [18] is not applicable to an elliptic arc. A model guided approach is used to locate and characterize ellipses.

Next step is to calculate the feature values between local features. These are minimum distance from the center of a circle to a line segment, minimum distance from the center of an ellipse to a line segment, the distance from the center of a circle to the center of an ellipse and the angle between a line segment and the major axis of an ellipse. Fig. 16 shows the results of local features superimposed over Fig. 14(a).

6. Representation of Objects by Volumes and Sweeps

From the modeling procedure described in section 2, objects are modeled by combining several simpler sub-parts. Instead of performing the set operations, volume properties and functional information can be attached to each of these parts and linked by Boolean expressions. Thus a CSG like volumetric mode is constructed from the CAD design.

Generalized cylinder (GC) is also an important representation in computer vision because it can concisely describe an important class of objects and its canonical frame can be efficiently used in matching. It is loosely characterized by having an axis, a set of cross-sections and a sweeping rule. The volume swept by these cross-sections along the axis using the given sweeping rule is the generalized cylinder. Here we use three steps to build a generalized cylinder model from the surface based CAD model: (1) Decompose the complex object into simpler sub-parts which can be represented as a single GC. (2) Extract the axis and cross sections for each sub-part. (3) Build junction relations between these GCs.

The ability to decompose a complex object into simple sub-parts plays an important role in human vision. The Recognition-By-Components (RBC) theory [6] explains the relation between the classic principles of perceptual organization and pattern recognition: the constraints toward regularization characterize not the complete object but the object's components. If an arrangement of two or more primitive components can be recovered from the input, objects can be quickly recognized even when they are occluded, rotated in depth, novel, or extensively degraded. There were several decomposition strategies: Nevatia [14] decomposed objects based on their axes and cross sections. Phillips et al. [16] decomposed objects using their compactness and Marr [13] decomposed 2-D images by connecting high concavity corners. Since the perceptual recognition of objects is conceptualized to be a process in which the image of the input is segmented at regions of deep concavity, the proposed decomposition strategy for 3-D objects is to decompose the object at its concave edges. This decomposition can also be done manually at the design stage.

To find the axis and cross sections from the surface, we use a recursive splitting method, and the idea of moment of inertia. The splitting method is similar to the iterative end point fitting technique used for curve approximation. The steps involved are:

1. Find the major axis of inertia, the one having minimum mement of inertia, and the extrema of the object along this axis.
2. Find the cross sections near these extrema which are perpendicular to the major axis of inertia and connect their centroids as the first approximation to the axis.
3. Find the cross section which passes through the mid-point of the approximated axis and is perpendicular to it. Connect its centroid to the two end points of the previous axis and split the axis and the object into two pieces.
4. Repeat step 3 on each of the sub-pieces recursively until the desired resolution is obtained or when the new cross section is not closed, i.e. it intersects with other cross sections.
5. Adjust the axis and cross sections recursively such that all cross sections are found at critical points of the axis and are perpendicular to the axis at their centroids.

This approach provides several advantages: (1) It uses the axis of inertia in the initialization procedure, which is invariant to rotation, translation and scale. (2) The axis is perpendicular to the cross sections and passes through their centroid. (3) The cross sections are closed planar curves and do not intersect each other. (2) and (3) minimize the number of possible GC representations for one object and achieve the uniqueness property of a vision model. (4) Since the whole surface is split during the recursion, its time complexity is improved to $O(m^*\log(n))$ where m is the total number of polygons in the CAD model of a component and n is the number of cross sections. Fig. 17 shows a deformed ellipsoid and the cross sections perpendicular to its principle axes. Fig. 18 shows the recursion of the axis and cross-section extraction procedure.

7. Conclusions

In this paper we presented four approaches for building 3-D models using CAD techniques which are suited to the problem of 3-D object recognition and manipulation. The efficiency of a representation depends upon the type of objects and the intended application. However, models based on multiple representations which include both surface and volume representations in a modular hierarchical manner provide all the advantages one can get. By using the CAD based model building approach such models can be built systematically. Work is in progress on the refinement of these models and using them in recognition strategies.

References

[1] H. Asada and M. Brady. The Curvature Primal Sketch. *IEEE Trans. on Pattern Analysis and Machine Intelligence* PAMI-8(1):2-14, January, 1986.

[2] P.J. Besl and R.C. Jain. Invariant Surface Characteristics for 3D Object Recognition in Range Images. *Computer Vision, Graphics, and Image Processing* 33:33-80, January, 1986.

[3] B. Bhanu and T. Henderson. CAGD Based 3-D Vision. In *International Conference on Robotics and Automation*, pages 411-417. IEEE, March, 1985.

[4] B. Bhanu and C.C. Ho. Computer-Aided Geometric Design Based 3-D Models for Machine Vision. In *Proc. of the 8th International Conference on Pattern Recognition.* Paris, France, October, 1986.

[5] B. Bhanu, S. Lee, C.C. Ho and T. Henderson. Range Data Processing: Representation of Surfaces by Edges. In *Proc. of the 8th International Conference on Pattern Recognition.* Paris, France, October, 1986.

[6] I. Biederman. Human Image Understanding: Recent Research and a Theory. *Computer Vision, Graphics, and Image Processing* 32:29-73, 1985.

[7] M. Brady, J. Ponce, A. Yuille, and H. Asada. Describing Surfaces. *Computer Vision, Graphics, and Image Processing* 32:1-28, 1985.

[8] 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.

[9] Carl de Boor. *Applied Math Series.* Volume 27: *A Practical Guide to Splines.* Springer Verlag, New York, 1978.

[10] I.D. Faux and M.J. Pratt. *Computational Geometry for Design and Manufacture.* John Wiley & Sons, 1979.

[11] R.M. Haralick, L.T. Watson, and T.J. Laffey. The Topographic Primal Sketch. *The Int. J. of Robotics Research* 2(1):50-72, Spring, 1983.

[12] B.K.P. Horn. Extended Gaussian Images. In *Proceedings of the IEEE*, pages 1671-1686. IEEE, December, 1984.

[13] D. Marr. Analysis of Occluding Contour. *Proc. Roy. Soc. London B* 197:441-475, 1977.

[14] R. Nevatia and T.O. Binford. Description and Recognition of Curved Objects. *Artificial Intelligence* 8:77-98, 1977.

[15] T. Pavlidis. *Algorithms for Graphics & Image Processing.* Computer Science Press, 1982.

[16] T. Phillips, R. Cannon, and A. Rosenfeld. Decomposition and Approximation of Three-Dimensional Solids. *Computer Vision, Graphics, and Image Processing* 33:307-317, March, 1986.

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

[18] S. Tsuji and F. Matsumoto. Detection of Ellipses by a Modified Hough Transformation. *IEEE Trans. on Computers* c-27(8):777-781, August, 1978.

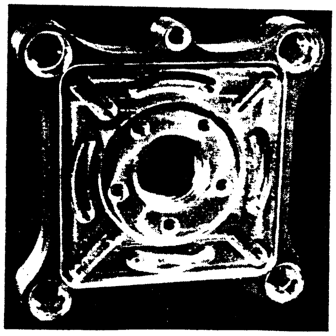


Fig. 1 Green Piece Object

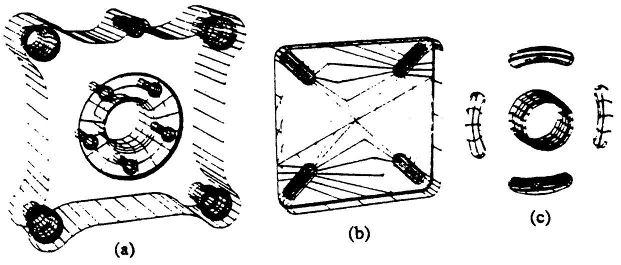


Fig. 2 Sub-parts for Green Piece CAD Model

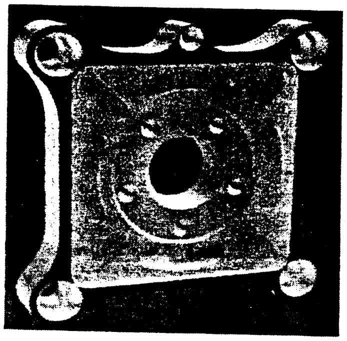


Fig. 3 Designed CAD Model for Green Piece

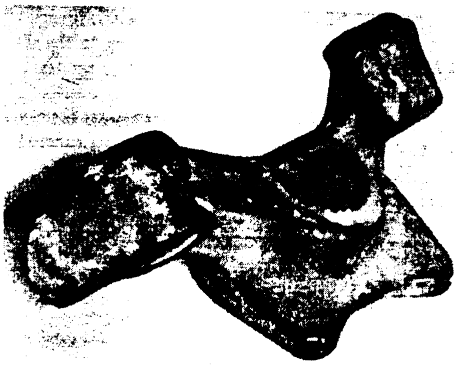


Fig. 4 Renault Piece Object

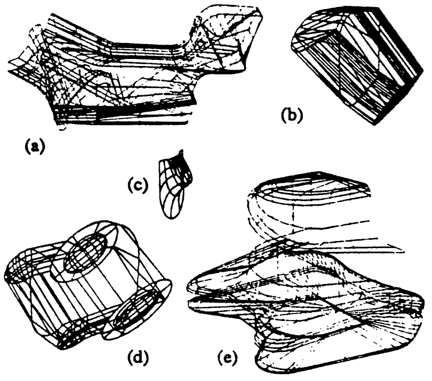


Fig. 5 Sub-parts for Renault Piece CAD Model

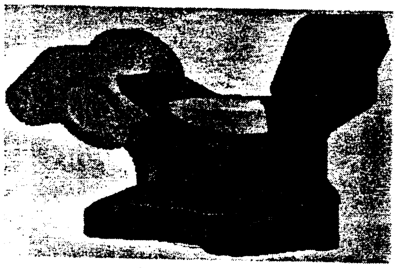


Fig. 7 Designed CAD Model for Renault Piece

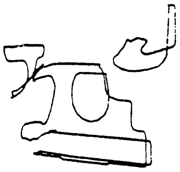
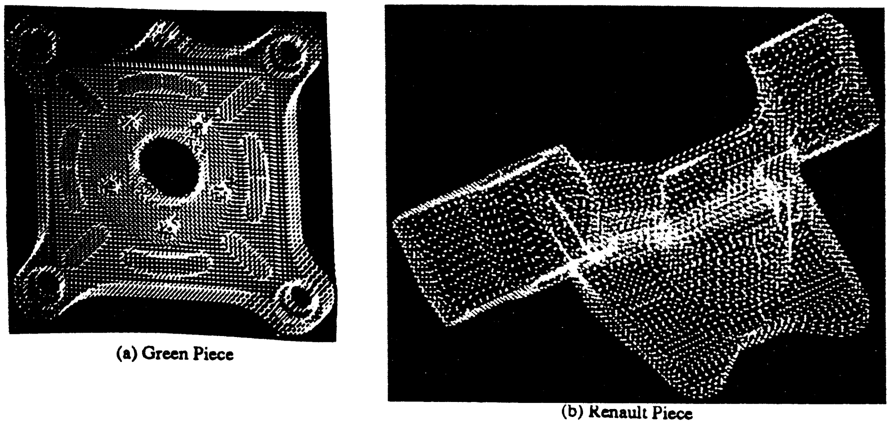


Fig. 6 Intersection Curves of Sub-parts of the Renault Piece


(a) Green Piece

(b) Renault Piece

Fig. 8 Surface Points Extracted From the CAD Model for Green Piece and Renault Piece, 0.2 Inch Resolution

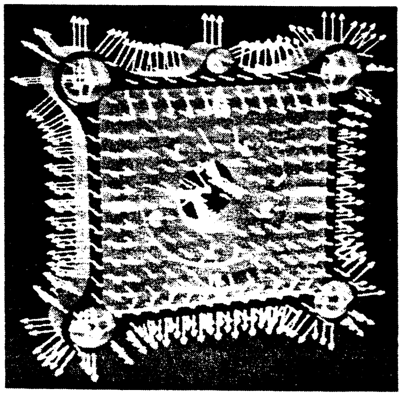
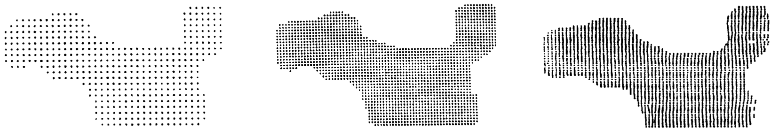


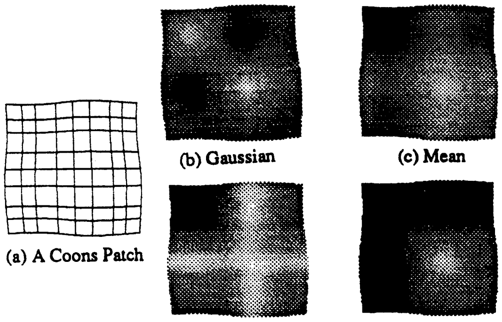
Fig. 9 Normals at the Surface Points for Green Piece, 0.4 Inch Resolution


(a) 0.2 Inch Spacing

(b) 0.1 Inch Spacing

(c) Real Range Data

Fig. 10 Synthetic Surface Points at Various Resolutions and Real Range Data of the Renault Piece


(a) A Coons Patch

(b) Gaussian

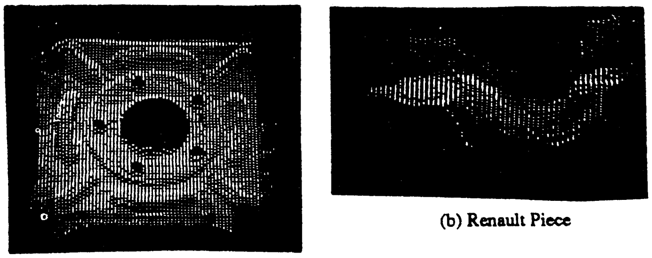
(c) Mean

(d) Maximum

(e) Minimum

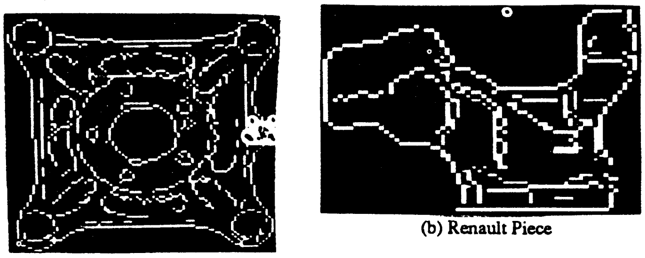
Fig. 11 Four Basic Types of Surface Curvature


Fig. 12 Edge Points as the High Curvature Points in Fig. 10


(a) Green Piece

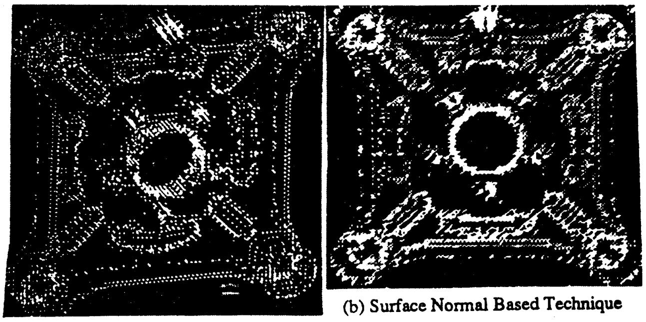
(b) Renault Piece

Fig. 13 Sensed Objects Used for Finding Edges


(a) Green Piece

(b) Renault Piece

Fig. 14 Linked Edges by Using the Gradient Technique


(a) Line Fitting Technique

(b) Surface Normal Based Technique

Fig. 15 Edge Points of Green Piece Using Different Techniques

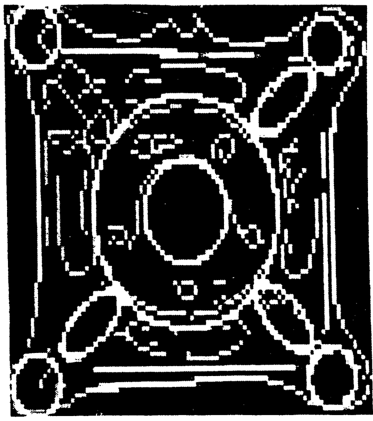


Fig. 16 Local Features(Lines, Circles, Ellipses) in Green Piece Range Image.

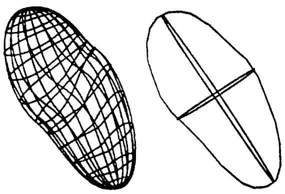


Fig. 17 A Deformed Ellipsoid and Cross Sections on Principal Axes

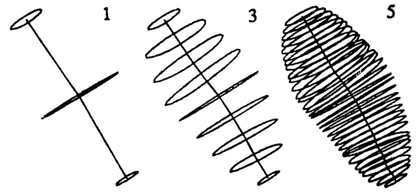
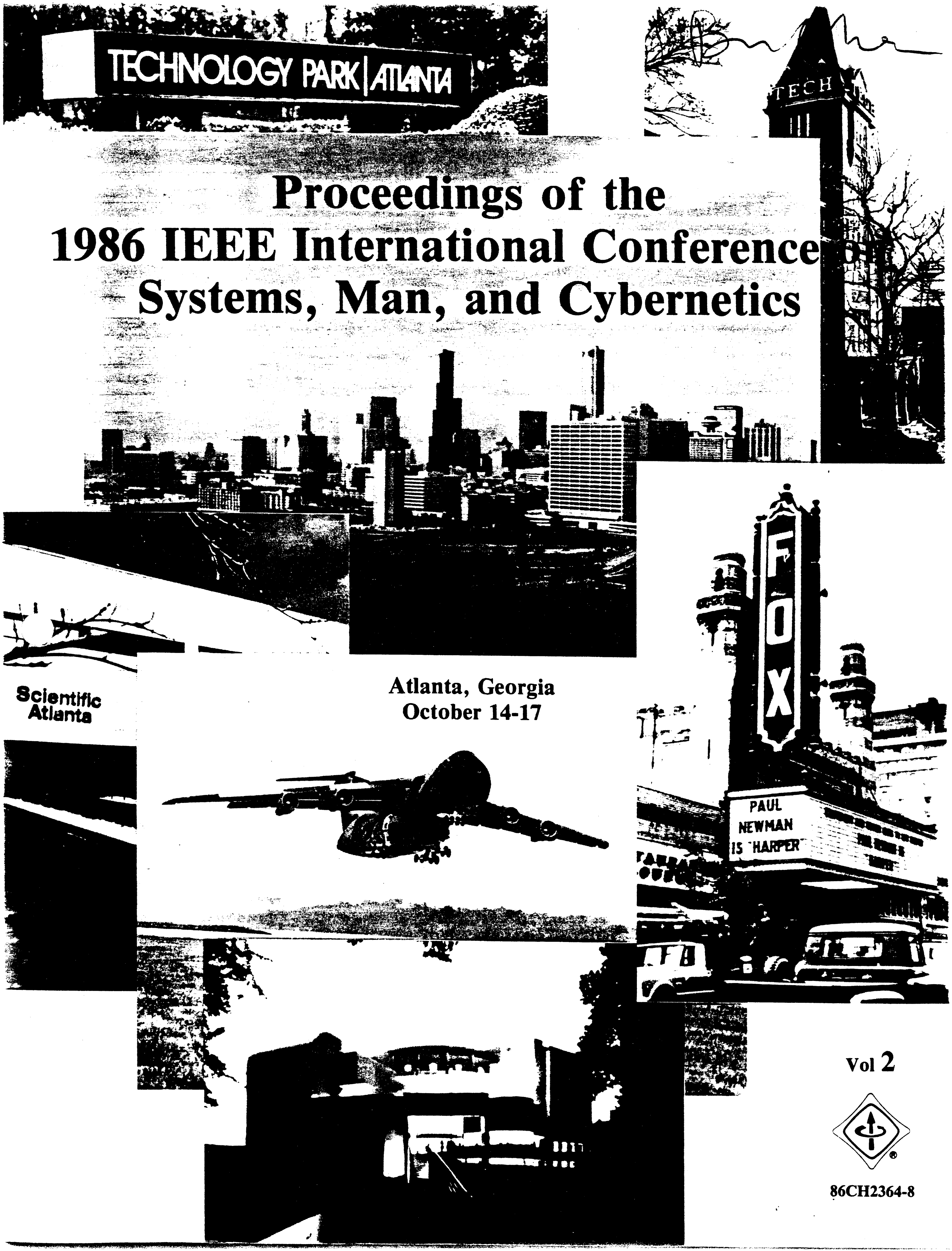


Fig. 18 Generized Cylinders After 1, 3, and 5 Recursions



Proceedings of the 1986 IEEE International Conference on Systems, Man, and Cybernetics

Atlanta, Georgia
October 14-17

Vol 2

86CH2364-8

Bir Bhanu

Proceedings of the 1986 IEEE International Conference on Systems, Man, and Cybernetics

Pierremont Plaza Hotel and Conference Center
Atlanta, Georgia
October 14-17, 1986

86CH2364-8