Building Hierarchical Vision Model of Objects with Multiple Representations

Bir Bhanu[+] and Chih-Cheng Ho[*]

+ College of Engineering, University of California, Riverside, CA 92521

* Department of Computer Science, University of Utah, Salt lake City, UT 84112

ABSTRACT

Building a hierarchical vision model of an object with multiple representations requires two steps: (1) decomposing the object into parts/subparts and obtaining appropriate representations, and (2) constructing relational links between decomposed parts/subparts obtained in step (1). In this paper, we describe volume-based decomposition and surface-based decomposition of 3-D objects into parts, where the objects are designed by a B-spline based geometric modeler called Alpha_1. Multiple-representation descriptions can be derived for each of these subparts using various techniques such as polygonal approximation, concave/convex edge detection, curvature extrema and surface normals. For example, subparts of a hammer can be described by two generalized cylinders or one generalized cylinder and one polyhedron. Several examples are presented.

1. INTRODUCTION

New developments in the integration of Computer-Aided Design and Manufacturing (CAD/CAM) allow both part design and manufacturing planning to take place concurrently. Certain new manufacturing processes, such as solid freeform fabrication, can even produce parts directly from 3-D CAD models. Computer vision can be applied in many automatic manufacturing tasks, such as inspection, assembly, robotics, etc. Most existing vision systems rely on models generated in an ad hoc manner and have no explicit relation to the CAD design, based on which the product was produced.

A CAD-based vision system, (see Figure 1), provides hierarchical vision models with multiple representations that are directly derived from the CAD database. The systematic approach, through decomposition and representation transformation, can automatically generate vision models for parts designed using CAD systems.[1] Part recognization strategies can also be studied and preplanned during the early stages of the design/manufacturing cycles, even before any actual part is manufactured. Multiple representation allows different matching strategies to be applied for the same object, or even for different parts of the same object.

2. RECOGNITION BY PARTS

The psychological theory of Recognition-By-Components (RBC), proposed by Biederman[3] provides a principal account of the perceptual organization and the pattern recognition.

Objects are recognized not by the concepts of the whole body but the object's components. If some of the parts of an object can be readily identified in their specified arrangement, the object identification will be fast and accurate. According to the studies by Tversky and Hemenway,[9] parts are the basic level of human concepts not only because smaller parts are easier to deal with, but also because different parts are to be handled differently. They proposed that part configuration underlies the various empirical operations of perception, behavior and communication that converge at the basic level. When describing or comprehending some body of knowledge or set of phenomena, humans often begin by decomposing the thing to be understood into separate parts. Through parts, humans use structure to comprehend, infer, and predict functions. This makes parts the most informative level.

Another importance of parts in early human conceptual development was given by Mervis and Greco.[6] Good examples help learning more rapidly and more accurately because they have more of the perceptually salient and functionally significant parts; these are also the parts most frequently shared by category members. On the contrary, poor examples that are less likely to share these attributes sometimes can even confuse the learning process.

3. OBJECT DECOMPOSITION

Division of objects into regular primitives (spheres, cubes, tetrahedra, etc.) is used in CSG (Constructive Solid Geometry) systems and in applications that use CSG representation. This is useful in CAD/CAM applications because of the analogy between set operations and mechanical manufacturing. However, this decomposition contains primitives that do not exist in the visual scene; thus, it is not suitable for computer vision applications.

Phillips et al. [8] decomposed complex objects like rocks into compact subparts. They used the convex enclosure to approximate the convex hull of an object, and computed the enclosure deficiency as a measure of the compactness of objects. They decomposed non-compact object recursively by a shrinking method until each subobject is sufficiently compact. By approximating each subpart with ellipsoid, they could compute properties of the object. However, this method will not work for object like torus, which has high enclosure deficiency but cannot be decomposed by shrinking. In fact, such an object is likely to be perceived as either a single object or an object with a big hole.

Nevatia and Binford[7] decomposed objects into generalized cylinders based on their axes and cross sections. They segmented an object into parts that can be described by "smooth" generalized cylinders, where axis directions and cross-section functions change continuously. First, they used a projection technique to determine local cones with axis direction pointing in a number of equally-spaced directions (typically 8) of an object. Second, they refined each local axis by a iterative axis/cross-section modification and merged consecutive cones by an extension method until a discontinuity or jump occurred. Third, they chose the longest or most elongated axis of each subpart. This segmentation is not expected to be perfect, in the sense in which human will segment it. Of course, it is not unique either, although the number of alternatives is small.

Marr[5] examined various types of joints between two generalized cylinders - side-to-end and end-to-end. He used segmented 2-D contours and assumed that the surface inside the enclosed contour is continuous. 2-D contours were decomposed by connecting high concavity corners. The result of this decomposition is similar to what human does. However, the heuristic rules used in connecting those strong segmentation points may not give a good decomposition in more complex cases.

Hoffman and Richards[4] proposed a minima rule for partitioning surfaces based on the generic intersection of surfaces - surfaces intersect transversally. The minima rule is to divide a surface into parts at loci of negative minima of each principal curvature along its associated family of lines of curvature. It will work only if the concave discontinuities form a closed contour. However, in this case the decomposition is not unique and different partitions are all reasonable.

4. OUR APPROACH

4.1 Introduction

The CAD system used in this work is the Alpha_1 solid modeling system,[10] developed at the University of Utah. Alpha_1 models the geometry of solid objects by representing their boundaries using NURBS (NonUniform Rational B-spline Surfaces). NURBS not only allow efficient representation for complex sculptured objects, but also give exact representation for simple primitives, such as sphere, cone, torus, etc. Mathematical properties, such as locality, continuity and variation diminishing, make NURBS a good design tool.

Using Alpha_1, objects can be designed with various geometric operators, such as extrude, bend, stretch, warp, etc., or combinations of them using boolean operations, set union, difference, intersection, etc. Trimmed NURBS, regions of a surface patch cut away are used to represent the results of boolean operations. The ability to perform set operations on sculptured surfaces, possibly nonclosed, combines the advantages of both CSG solid modeling and boundary representation modeling. Alpha_1 also provides high quality graphics and feature-based mechanical design and manufacturing, as well as a testbed for many advanced research ideas.

In this work, models are decomposed along their concave edges. It is similar to the minima rule by Hoffman and Richards and is a 3-D extension of Marr's method. B-spline models are approximated by polyhedra. Edge detection using surface normal curvature, approximated by surface normal vectors, marks every possible edge as convex or concave and sharp or rounded. Finally, surfaces are separated by enclosing them with concave edges.

4.2 Polygonal Approximation

For systems that create polyhedral CAD models, their models can be adopted directly in the vision analysis. However, in Alpha_1 B-spline surfaces are used as the only underlying structure. Every surface, curved or planar, is described as tensor product B-splines. It is therefore, required to approximate the surface by polygons in order to obtain a polyhedral description.

Subdivisions are first applied to curved surfaces in order to obtain flat patches, within a given tolerance. Polygonal approximations of flat patches then result in a polyhedral approximation of the B-spline CAD model. Two steps are needed to have a valid polyhedral model: First, the original CAD model should be valid and meet a set of geometrical and topological conditions. Second, the subdivision and polygonal approximation procedures should preserve the validation conditions. To satisfy these conditions, constraints are added to the Alpha_1 CAD designs to make sure that adjacent patches have compatible, not necessarily the same, parametrization along the common boundaries. The compatibility condition is maintained easily by constructing adjacent patches from direct derivations (refinements, concatenations, extractions, etc.) of the same B-spline curve along the adjacent common boundaries. Also, adjacency information on surface patches contains not only the adjacent patches and their adjacent sides but also the portion of the boundaries along which they are matched. For partially or multiple adjacent patches, more than one adjacency information is asserted on each side. This information is propagated to the subdivided patches whenever a subdivision occurs.

A global approach to the polygonal approximation is used such that all the required subdivisions are first performed and then polygons are built for every small subpatch. Each polygon contains not only the subpatch's four corners but also the adjacent corners of all neighboring patches. Therefore, the adjacency information on the subpatch can be mapped onto each side of the approximated polygon thereby maintaining the topological validity of the resulting polyhedron.

Figure 2(b) shows a polyhedral approximation for the airplane CAD model in Figure 2(a). Figure 3(a) shows a B-spline CAD model for a teapot. Its polyhedral approximation is shown in Figure 3(b). Features can also be extracted from this polyhedral representation. For example, Figure 4 shows the results of edge detection on the airplane and the teapot models by thresholding the changes of the surface normal vectors along adjacent faces. Local features of edges (such as line segments, circles, ellipses, etc.) can also be extracted from these results.[2]

4.3 Edge Detection

In general, edges can be defined as the contours of discontinuity of image/model properties. Similarly, regions can be defined as the areas of continuity of properties. For example, in 2D image analysis, edge pixels are characterized by having large intensity gradients compared to their neighbors. Regions are the collection of pixels whose intensity are either close to each other, or changing gradually. Edges can also be defined by the boundaries of regions, and regions as the interior of edge contours. However, edges extracted as pixels of high intensity gradient may not always form closed contours. Sometimes, zero crossings are used to form closed regions.

For 3D polyhedral objects, the property that we use is the surface normal. We define edges to be the discontinuity, or sharp directional changes, of surface normal. For curved objects, this is equivalent to the points of high surface curvature. For example, Figure 5 shows edge detection on surface points sampling at various resolutions for the Renault piece by thresholding the surface principal curvatures. The rightmost figure in Figure 5(a) shows the range data acquired from a laser range finder with 0.12x0.08 inch spacing. The other two are sampled from the CAD model with 0.2x0.2 and 0.1x0.1 spacing respectively. Figure 6 shows Gaussian curvature and the principal curvatures of the teapot CAD model. Figure 7 shows the zero crossings of Gaussian curvature. It is noted that the zero crossings not only give the edges of high curvature and step edges, but also the edges corresponding to points of inflection. Points of inflection can further segment faces into concave, convex, or saddle regions.

4.4 Decomposition

Regions, usually called faces in 3D, are the connected area where surface normals are either constant (flat faces) or change slowly (curved faces). Region growing on 3D surface is similar to 2D region growing problem. From a polyhedral approximation of the CAD model, edges are detected and marked on each side of a face. To form a region bounded by the detected edges, we do a depth-first traversal starting from an unvisited subdivision patch. For each neighboring patch, if the neighbor is not visited and the shared boundary between them is not marked as an edge, we mark the neighbor as visited and it is included in the same region as the current patch. Otherwise, the growing is stopped and we go back to the next neighbor. When the traversal ends and back to the starting patch, we have collected all the patches of the region containing the staring patch. We repeat the region growing process from another unvisited patch until all patches are visited and all regions are found.

4.4.1 Volume-Based Decomposition

To decompose a 3D object into solid parts, models are decomposed along the concave edges. This method is similar to the minima rule by Hoffman and Richards, and is a 3D extension of Marr's method (see section 3). As stated above, edges are detected by using surface curvatures, or are obtained by using surface normals. Concave edges are those with negative Gaussian curvature. Surface normal can also be used to detect whether an edge is concave or convex. For true polyhedral objects, where edges and faces are all linear order, the latter will be used.

Volume-based decomposition involves identifying the concave edges along which the decomposition occurs. If we connect two points, each from the interior of one of the two adjacent faces, the line segment between these two points should all fall outside the object if the edge between the faces is concave, and inside the object if the edge is convex. In other words, if a point from face1 is outside the tangent plane of face2, the edge between face1 and face2 is a concave edge, provided face1 is convex. For general cases, concave or curved faces, the selected point must be close to the edge, preferably somewhere in the middle, but not on the edge.

Figure 8 shows a simplified model of a hammer and the result of edge detection on the polyhedral approximation of it. The solid lines are the convex edges, and the dashed line is a concave edge. Figure 9 shows two subparts of the hammer from the volume-based decomposition. Decomposition of an airplane model (Figure 2(b)) is shown in Figure 10 and generalized cylinder representation is shown in Figure 11.

4.4.2 Surface-Based Decomposition

The decomposition of objects into solid parts shows interesting results and can be useful in applications such as assembly. For computer vision applications, since the only thing we see is the surface of objects, surface-based decomposition is useful and necessary, in addition to volume-based decomposition.

To decompose an object into faces, we simply include the convex edges in the region growing process. Figure 12 shows the result of decomposing the hammer model into faces. If we include all zero crossings of Gaussian curvature, the decomposed faces can also be labeled as flat, convex, concave, or saddle. By using the two principal surface curvatures, surface can be decomposed further into cylindrical, spherical patches, etc.

4.5 Vision Model Having Multiple Representations

A CAD system may use a representation scheme, such as NURBS or CSG tree, that is good for modeling but does not, in general, contain all the features that are important for computer vision applications. A systematic approach to building vision models is to construct features from the CAD database and incorporate those in the vision models that are crucial for object recognition and manipulation.

Most existing vision systems use only one representation in their models. However, there is no single representation or a matching technique based on a single representation that can efficiently and reliably represent different classes of 3D objects for object recognition. The CAD-based approach allows the construction of models employing several representations; thus, is able to handle a wider class of objects. Moreover, it allows different parts of the same object to have different representations.

Multiple-representations description can be derived for each of the subparts using various techniques described by Bhanu and Ho.[1] For example, subparts of the hammer can be described as two GCs or one GC and one polyhedron (see Figure 13).

5. CONCLUSIONS

In this paper we have presented approaches for building hierarchical vision models of objects with multiple representations. From a computational point of view, hierarchical representation simplifies the complexity of the problem. Representations of a 3-D object using hierarchical structures based on the decomposition of the object's surface and/or volume are helpful in many practical situations. For computer vision applications, such as object recognition, it provides a solution for the recognition of partially occluded objects. This is important since self occlusion may occur even for a single object, specially if the object is concave. Psychological studies have demonstrated the importance of object parts in human visual recognition. We are using multiple object representations in our current work for object recognition.

REFERENCES

1. B. Bhanu and C.C. Ho, "CAD-Based 3-D Object Representations for Robot Vision," *IEEE Computer* **20**(8) pp. 19-35 (August 1987).

2. B. Bhanu, C.C. Ho, and S. Lee, "CAD Based 3-D Models for Computer Vision," The International Conference on Systems, Man and Cybernetics, Atlanta, Georgia, pp. 1002-1006 (October 1986).

3. I. Biederman, "Human Image Understanding: Recent Research and a Theory," *Computer Vision, Graphics and Image Processing* **32** pp. 29-73 (1985).

4. D. D. Hoffman and W. Richards, "Parts of Recognition," *Cognition* **18** pp. 65-96 (1984).

5. D. Marr, "Analysis of Occluding Contour," *Proc. Royal Soc. London B* **197** pp. 441-475 (1977).

6. C.B. Mervis and C. Greco, "Parts and Early Conceptual Development: Comment on Tversky and Hemenway," *Journal of Experimental Psychology: General* **113**(2) pp. 194-197 (June 1984).

7. R. Nevatia and T. Binford, "Description and Recognition of Curved Objects," *Artificial Intelligence* **8** pp. 77-98 (1977).

8. T. Phillips, R. Cannon, and A. Rosenfeld, "Decomposition and Approximation of Three-Dimensional Solids," *Computer Vision, Graphics and Image Processing* **33** pp. 307-317 (March 1986).

9. B. Tversky and K. Hemenway, "Objects, Parts, and Categories," *Journal of Experimental Psychology: General* **113**(2) pp. 169-193 (June 1984).

10. Alpha_1 Research Group, Department of Computer Science, University of Utah, "Alpha_1 Users Manual," *Salt Lake, City, Utah*, (1986).

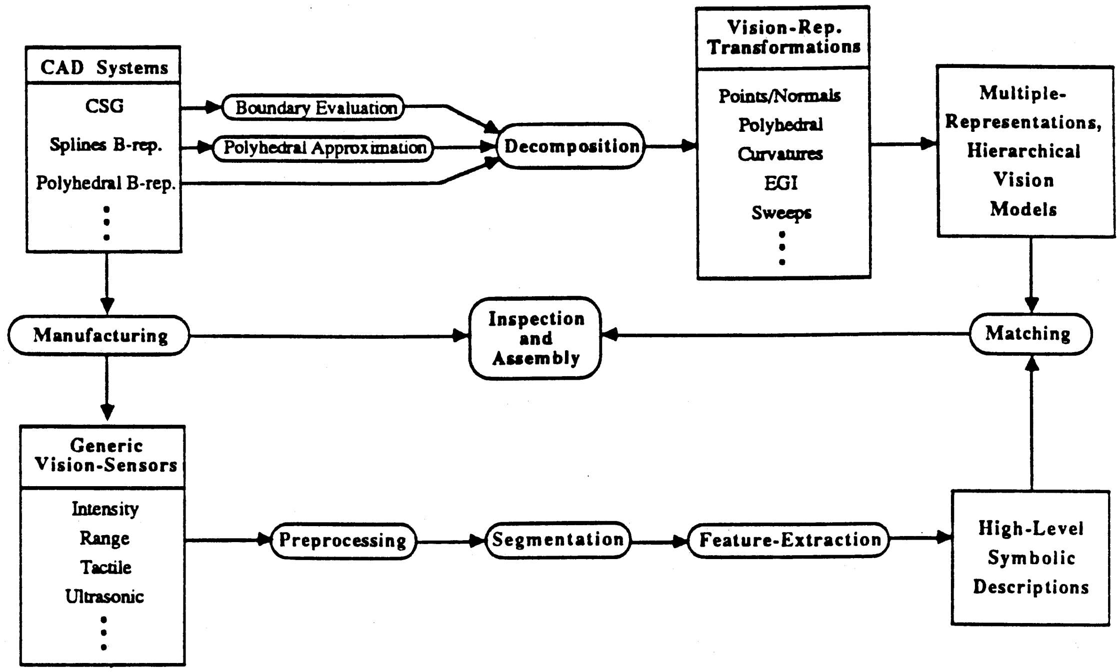
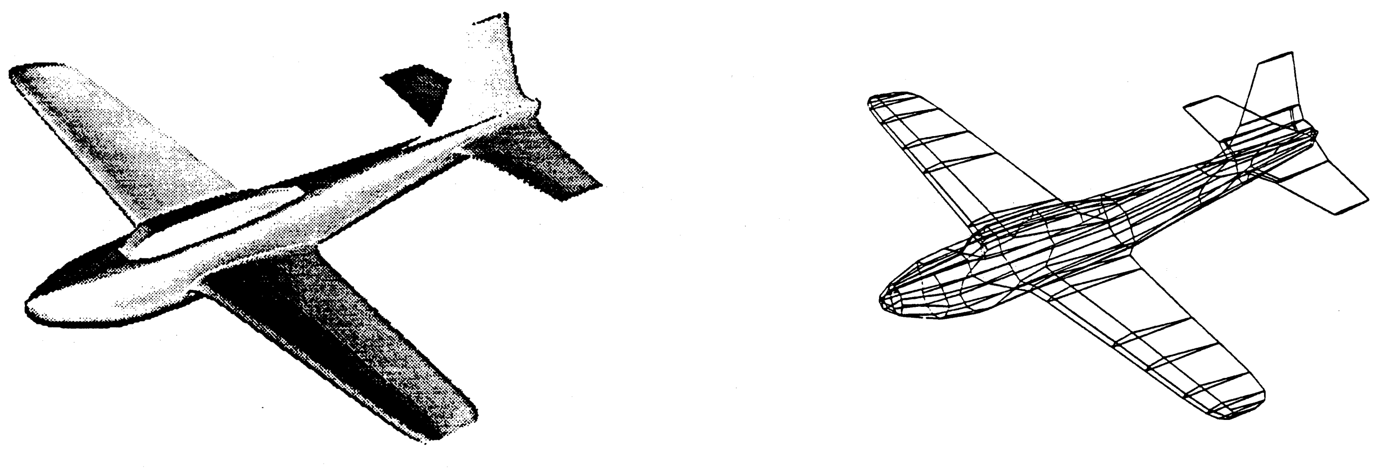
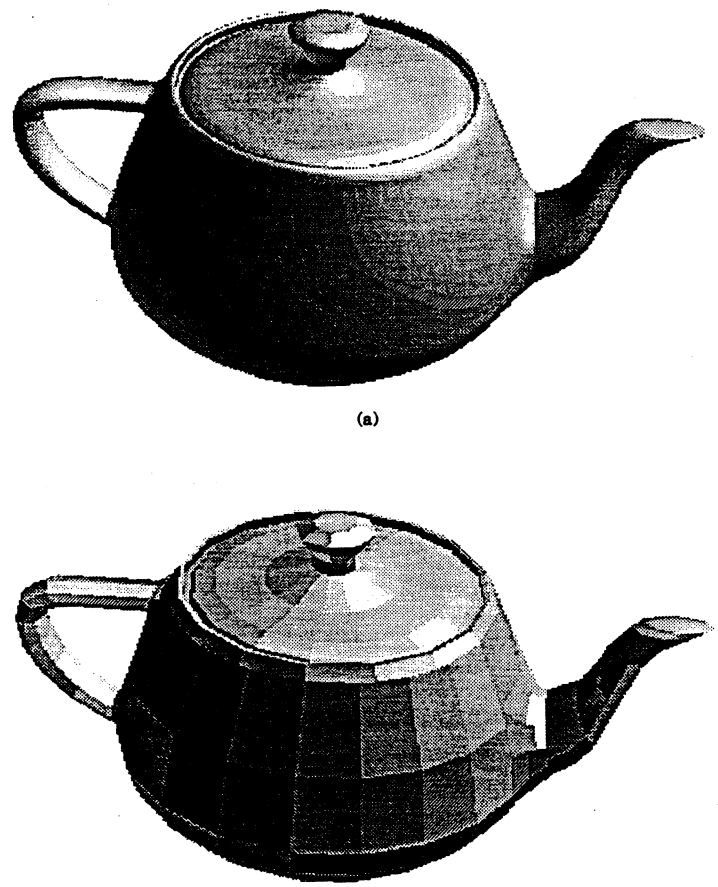


Figure 1. A generalized CAD-based vision system



(a) (b)

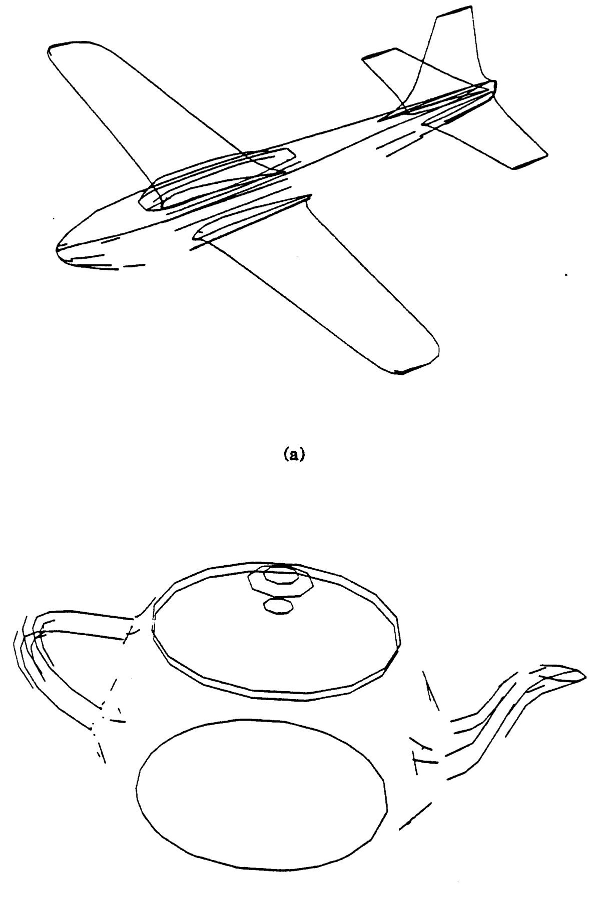
Figure 2. Airplane model
(a) B-spline model of an airplane
(b) Polyhedral approximation of (a)



(a)

(b)

Figure 3. Polyhedral representation for a teapot
(a) B-spline model for a teapot
(b) Polyhedral approximation of (a)

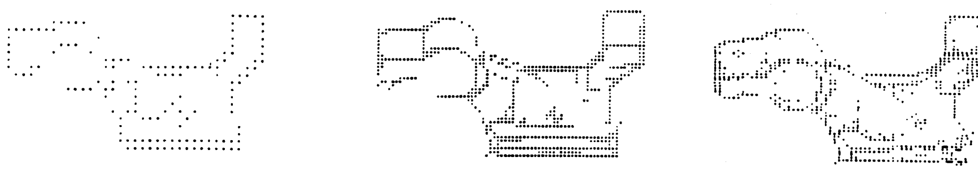


(a)

(b)

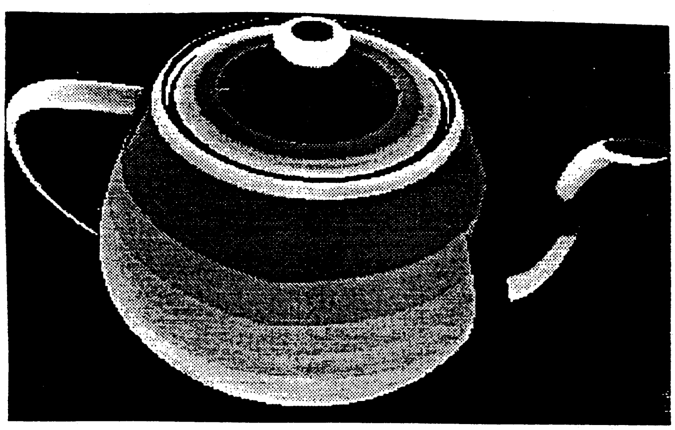
Figure 4. Extraction of features from the polyhedral approximation
(a) Edge detection on an airplane model
(b) Edge detection on a teapot model

(a)



(b)

Figure 5. Surface points and extrema of principal curvatures for Renault piece
(a) Sampling of surface points at various resolutions (from left to right: 0.2 x 0.2, 0.1 x 0.1, and 0.12 x 0.08 inch spacing),
(b) Edge points as the extrema of principal curvatures for figures in (a).



(a)

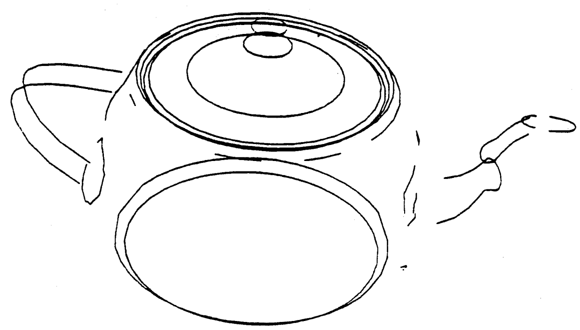
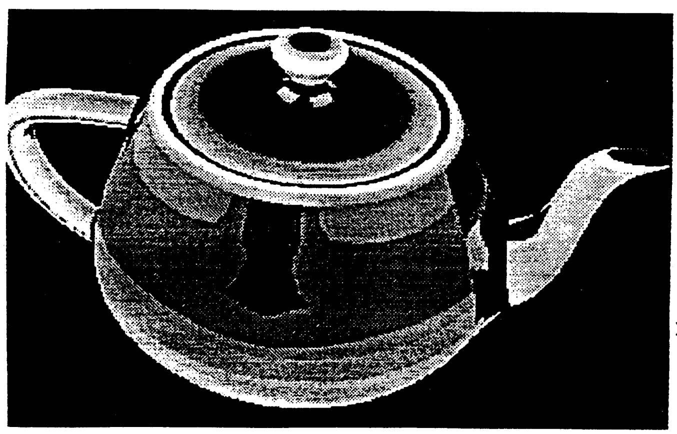
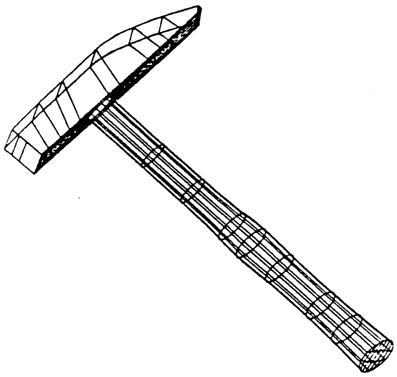


Figure 7. Zero crossings of Gaussian curvature on a teapot model



(b)

Figure 6. Surface curvatures of B-spline CAD models
(a) Gaussian curvature of teapot model
(b) The extrema of principal curvatures of a teapot model



(a)

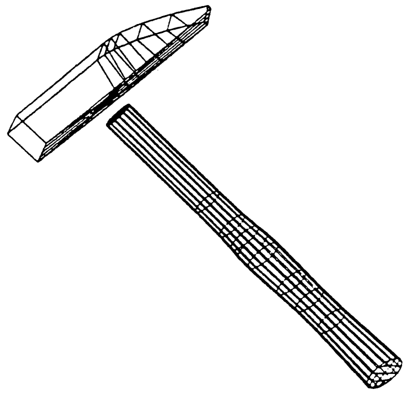
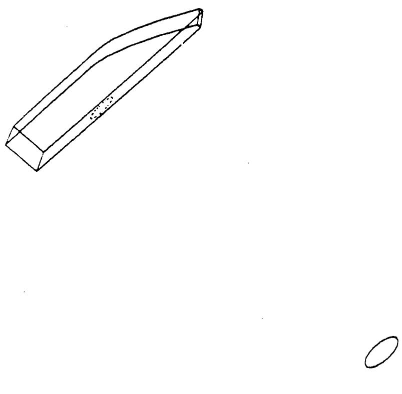
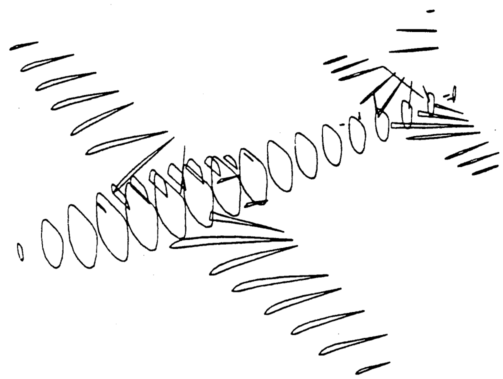


Figure 9. Decomposition of a hammer model

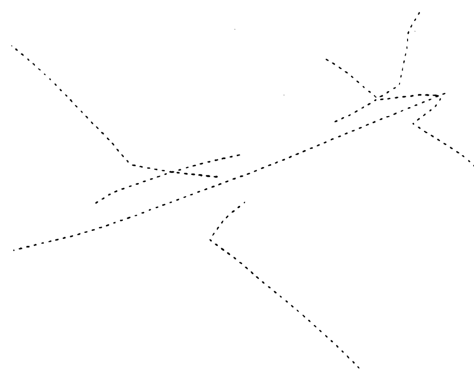


(b)

Figure 8. A hammer model and edge detection results
(a) A simplified hammer model
(b) Edge detection results on (a)



(a)



(b)

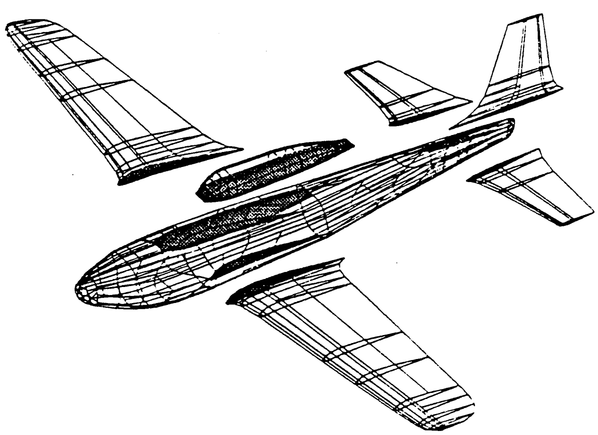


Figure 10. Decomposition of an airplane

Figure 11. Generalized cylinder representation on subparts of an airplane
(a) Cross sections of GCs of an airplane
(b) Axis of generalized cylinder in (a)

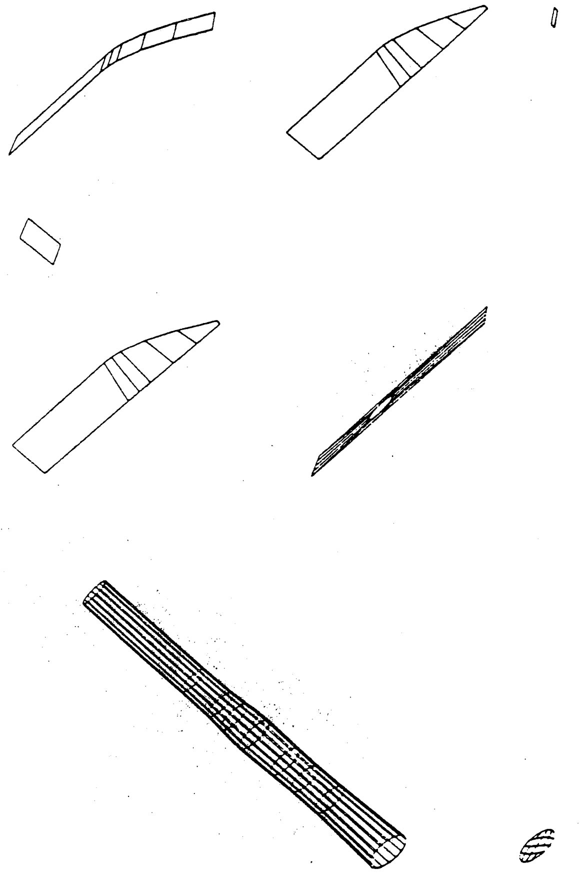
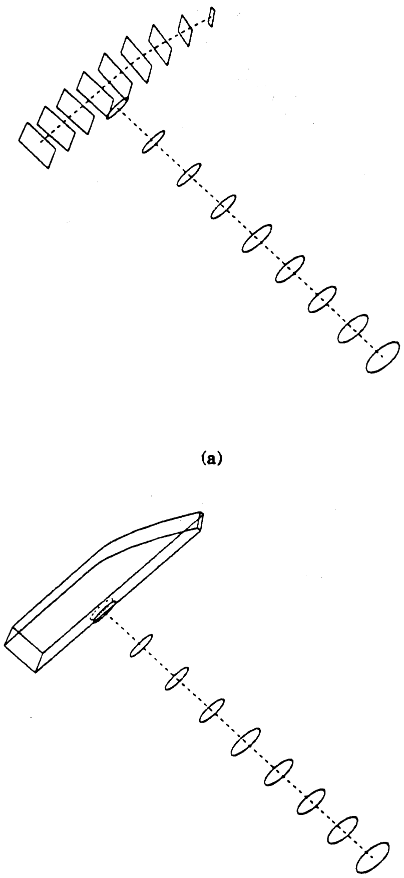


Figure 12. Surface decomposition into faces on a hammer model



(a)

(b)

Figure 13. Multiple representations of a hammer
(a) Two generalized cylinders representation
(b) One generalized cylinder and one polyhedron