3D Manufacturing Primitive CAD-Based Object Recognition

Leda Villalobos
Mechanical and Industrial Eng. Dept.
The University of Texas at El Paso
El Paso, TX 79968-0521
leda@nasa2.ece.utep.edu

Francis L. Merat
Electrical Engineering Dept.
Case Western Reserve University
Cleveland, OH 44106-7221
merat@pris.eeap.cwru.edu

ABSTRACT

This paper describes a 3-D recognition system for objects designed with a manufacturing primitive-based CAD. Within the recognition system, objects are modelled as organized compositions of instantiated manufacturing primitives. Since these primitives have a higher semantic level than the traditional entities used in object modeling (surfaces, edges, and so on), the burden of recognition is shifted from matching the scene to the objects database to matching the scene to the primitives database, which is usually significantly smaller. Consequently, set up time is reduced while recognition is faster. The paper reviews our approach to primitive and object modeling, primitive indexing, and pose estimation.

1. Introduction

The growing interest in concurrent engineering and its related areas has made an impact on the philosophy of object recognition for manufacturing. This is so because it has become apparent that vision systems for assembly and inspection can be constructed faster and maintained more efficiently by exploiting the geometry rich CAD models of the parts being manufactured [1, 2].

This paper presents an ongoing research effort to develop a 3-D object recognition system for parts designed with a *manufacturing primitive-based* CAD, a type of CAD system which preserves design intent throughout the different phases of manufacturing, thus reducing costs and lead time [3, 4, 5].

The heart of the approach resides in modeling objects as collections of inter-related manufacturing primitives. This allows the main object recognition task to be broken down into less stringent ones of primitive identification, instantiation, and pose estimation, followed by the actual object model indexing.

In section 2, we present the primitives currently supported by the recognition system, and describe our modeling schemes for both manufacturing primitives and objects. Primitive hypotheses generation is briefly described in Section 3, while verification and geometric instantiation are explained in Section 4. Primitive pose estimation is presented in Section 5. Finally, work-in-progress on part model indexing and pose estimation is presented in Section 6, with conclusions in Section 7.

2. 3D Primitive and Part Modeling

In this research, objects are modeled as set of instantiated manufacturing primitives. This section presents the schemes used to represent both the primitives and the parts constructed with them.

2.1. Manufacturing Primitives

Primitive representation is examined in the context of a reduced set of manufacturing primitives shown in Figure 1.

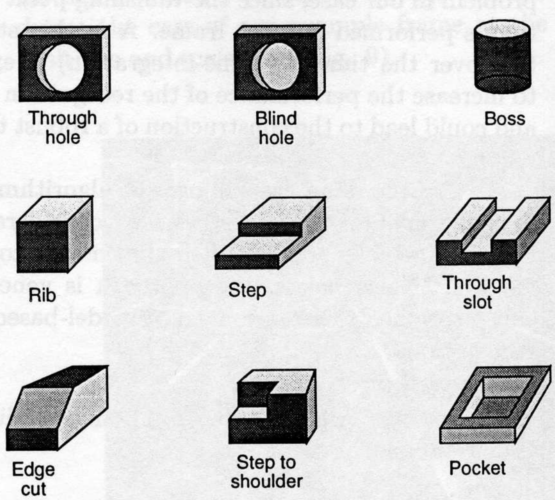


FIGURE 1. The CAD manufacturing primitives.

Nine manufacturing primitives were chosen: through hole, blind hole, boss, rib, through slot, step, step-to-shoulder, pocket, and edge cut. Their variety is sufficient for designing a substantial number of different parts, thus validating recognition methodology; nevertheless, they are simple enough to facilitate implementation and testing.

Primitives are modeled at two levels: topological and geometrical.

2.2. Primitive Topological Modeling

We take a viewer-centered approach to represent the primitives. This representation is, nevertheless, topological in nature. Contrary to most vision systems which restrict aspect views —or aspects, for short— to be purely edge constructions, our approach introduces a topological representation graph for each primitive aspect. An aspect is *topologically* described by a combination of surfaces with specific unary and binary relational properties.

The only unary feature included is the type (planar, cylindrical, and so on) of the surface patch. To generate the topological graph representation of an aspect, three binary relations are extracted for every pair of surfaces: the angle between their orientation vectors (parallel, perpendicular, or oblique); their spatial proximity (physically adjacent or not, and concavity/convexity of shared boundary); and their geometric equivalence [9]. A summary of the topological features which can exist between any two surfaces for the selected primitives is given in TABLE 1.

TABLE 1. Topological Features

FEATURE	Orientation	Adjacency	Equivalence
LEC	Parallel	Convex	No
LIC	Parallel	Concave	No
LID	Parallel	No	No
PEC	Perpendicular	Concave	No
PIC	Perpendicular	Concave	No
PID	Perpendicular	No	No
ANC	Oblique	Concave	No
PSE	Parallel	No	Yes

Each primitive has several aspects which we have numbered for identification purposes. For example, a blind hole has three aspects. Figure 2 depicts "aspect 1" of a blind hole and the corresponding surface topological graph.

FIGURE 2. One of the aspects and topological graphs for a blind hole. Thick, dark links in the graph indicate a LIC topological relation; thick, light links indicate a LEC relation; and thick, dashed links indicate a LID relation.

It should be emphasized that the simplicity of the selected primitives makes compact topological graphs possible. For primitives involving complex surface parametrization, such as splines, the representation of topological properties would require more elaborate relational trees [10].

2.3. Primitive Geometric Modeling

Primitive modeling is completed by adding a geometric level to describe surface boundaries. As explained later, this geometric information is used as a tool to verify/reject primitive hypotheses. The geometric descriptions include the types of surfaces in a primitive (planar, cylindrical, etc.), and their pertinent edges (jump, straight, circular arcs, etc.)

2.4. Part Modeling

In the part database, each part is described as an organized aggregate of instantiated manufacturing primitives. A primitive is instantiated by indicating the dimensions of its characteristic parameters, and its location in the part with respect to a global reference frame.

Characteristic parameters are instantiated following a convention for each primitive, such as the ones shown in Figure 3. For example, a through hole is characterized by its radius and the orientation of its axis. The Z axis of the local frame runs along the axis of the hole, pointing out of the material, while the origin is located at the opening of the hole.

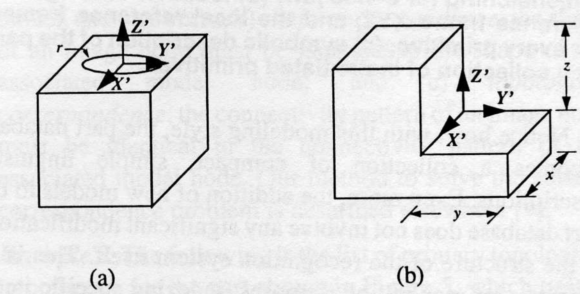
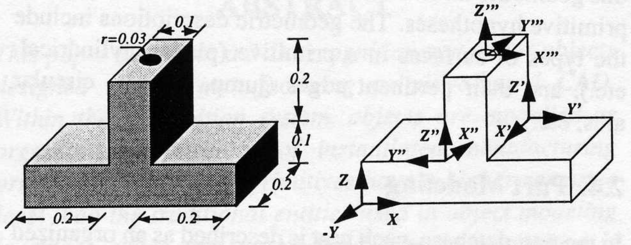


FIGURE 3. Primitive geometric parameters and definition of local reference frames. (a) A through hole is completely characterized by its radius and the axis orientation. (b) The geometry of the step is defined by the dimensions of the cut.

A primitive's pose is instantiated by first selecting a global reference frame which remains available throughout the fabrication of the part. Normally, this frame has already been established by the CAD system. The primitive's pose is given by the rotation matrix between the global and local reference frames, and the position vector which goes from the origin of the global reference frame to the origin of the local reference frame.

The model of a part has four slots. The slots indicate the part's name and its ID number, the dimensions of the raw stock from which the part is manufactured, the types of primitives present, and their complete instantiation (including geometric parameters and pose). Figure 4 shows the model of a vise jaw created with two steps and a through hole. This model was automatically obtained from the output of a manufacturing primitive-based CAD system [6].



(a) (b)

```
model(part_id(1,vise_jaw),
    stock(0.5,0.2,0.3),
    primitives([step,th_hole]),
    description([step(2, [(1,param([0.2,0.2,0.2]),
                pose([0,1,0],[-1,1,0],[0,0,1],[0.3,0.2,0.1])),
            (2,param([0.2,0.2,0.2]),
                pose([0,-1,0],[1,0,0],[0,0,1],[0.2,0,0.1])])]),
    th_hole(1, [(1,param([0.03]),
                pose([1,0,0],[0,1,0],[0,0,1],[0.25,0.1,0.3])])])])).
```

(c)

FIGURE 4. A part and its primitive description. (a) Dimensioning for a vise jaw; (b) location of the global reference frame *XYZ*, and the local reference frames for every primitive; (c) symbolic description of the part as a collection of instantiated primitives.

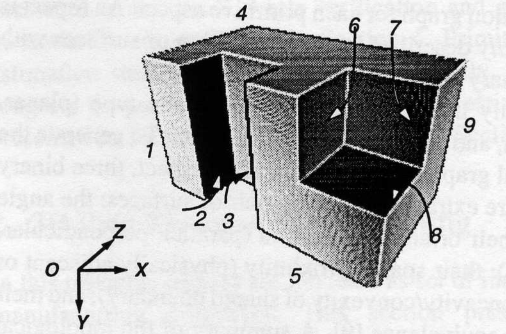
Notice how, with this modeling style, the part database becomes a collection of compact, simple linguistic descriptions. Even more, the addition of new models to the part database does not involve any significant modifications to the structure of the recognition system itself. This is so because, for recognition purposes, indexing specific items from the part database first requires the identification of the primitives present in the image. As long as the set of primitives remains unchanged, minimum modifications have to be implemented in the recognition system. Primitive indexing is described in the following sections.

3. Topological Primitive Indexing

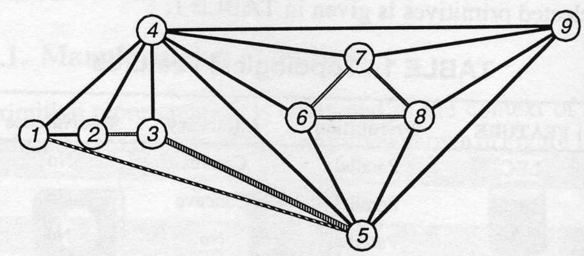
3.1. Topological Graphs

An undirected, attributed graph G is used to describe the topological relations between the surfaces of a segmented image [7]. Each surface in the segmented image is assigned a node in the graph, and for any two adjacent surfaces, there is a link connecting their corresponding nodes. Nodes are labeled according to the type of the associated surface, while links are labeled according to the topological relations of the surfaces interactions. The dictionary of topological relations is the same as that introduced in TABLE 1.

Example 1. Figure 5 shows the topological graph constructed from the segmented image of a simple part.



(a)



(b)

```
graph([(node(1),plane,links([(2,pec),(4,pec),(5,pse)])),
    (node(2),plane,links([(1,pec),(3,pic),(4,pec)])),
        ooo
    (node(9),plane,links([(4,pec),(5,pec),(7,pec),(8,pec)]))]).
```

(c)

FIGURE 5. Constructing the topological graph of a segmented image. (a) Image with individual surfaces identified; (b) corresponding topological graph; (c) portion of symbolic representation as a PROLOG clause.

3.2. Preliminary Topological Indexing

The goal of the preliminary hypotheses generation paradigm is to produce a coverture P of G, $P=\{P_j, j=1, 2, ..., K\}$, where P_j is a path of G for which there is supporting topological evidence that it could correspond to an instance of a primitive. Each path P_j has associated two sets of hypotheses, H_{cj} and H_{pj}. H_{cj} is the set of *complete aspect* *hypotheses*, while H_{pj} is the set of *partial aspect hypotheses* for path P_j. Complete aspect hypotheses are those primitive instances for which P_j could be a topological isomorphism; partial aspect hypotheses are those primitive instances for which P_j could be a topological sub-isomorphism.

The generation of hypotheses from topological information is done by first finding all paths in the topological image graph which are compatible with the form primitives. Valid paths are created with a breath first search on the connectivity of each node in the image graph. The search in each branch of the tree is pruned whenever the corresponding qualitative connectivity graph does not match any primitive pattern.

To accelerate topological indexing, we perform a qualitative pattern matching between the image and the primitives aspect graphs. The pattern has a total of 11 features organized in two groups, one related with the surface type description, and the other with the topological relations among the surfaces. The surface type portion of the pattern has 3 entries, indicating the number of planar, convex, and concave surfaces in the path, respectively. The topological description pattern has 8 features, which represent the sum of the graph's links with the topological relations shown in TABLE 1.

3.3. Topological Hypotheses Organization

Based on heuristics explained elsewhere [9], we have grouped the preliminary hypotheses in three levels. The highest level of the hierarchy includes those paths which are not subsets of other paths, and have a non-empty set of complete aspect hypotheses. The next level includes those paths which are not subsets of other paths, and have a non-empty set of partial aspect hypotheses. Notice that it is possible to find the same path in both the first and second hierarchical levels. The last level of the hierarchy includes all remaining preliminary hypotheses paths. We refer to the first level of the hierarchy as the *primary topological hypotheses*, to the second as the *secondary topological hypotheses*, and to the third as the *tertiary topological hypotheses*. The third level is primarily used to propose solutions when there is substantial destructive occlusion.

Once hypotheses are divided, they are pruned. Not all hypotheses are tested, though. Initially, only primary hypotheses examined. If the evaluation of these hypotheses is not sufficient to fully explain the scene, selected secondary and tertiary hypotheses are also evaluated.

Example 2. The following is the list of primary topological hypotheses generated for the part depicted in Figure 5.

```
Primary_topological_hypotheses = [path([2,4,6,8], [s6]),
                                  path([1,4,6,8], [s6]),
                                  path([3,4,6,8], [s6]),
                                  path([2,4,7,8], [s6]),
                                  path([1,4,7,8], [s6]),
                                  path([3,4,7,8], [s6]),
                                  path([1,2,3,4,5], [ts8]),
                                  path([4,5,6,7,8,9], [ss2])]
```

Notice that, since they are generated through a qualitative matching, some of the hypotheses are incorrect. As it will be seen later, the false hypotheses are pruned once the connectivity patterns of each node in the surface connectivity graph are analyzed. The correct hypotheses for the step-to-shoulder (aspect 2) and the through slot (aspect 8), are also generated.

3.4. Hypotheses Pruning

Pruning is performed by matching the isomorphism or sub-isomorphism of the topological graph of the path with respect to the graph of the hypothesized primitive instance. The purpose of this matching is twofold. First, success/failure at matching the graphs further prunes the set of hypotheses; second, should the matching be successful, the solution of the surface correspondence problem would greatly simplify future hypotheses verification through surface fitting or edge tracing, since it becomes a model driven task.

The desired solution to our topological isomorphism problem requires the satisfaction of three types of constraints: a) *uniqueness correspondence*; each surface in the image eventually matches only one surface in the model, and vice versa; b) *shape correspondence*; the shape of an image node must be identical to the shape of the associated model node; and c) *topological correspondence*; the connectivity pattern of an image node must be identical to the connectivity pattern of the associated model node. Our method to solve the surface correspondence problem is described elsewhere [9].

Example 3. The following is the list of primary topological hypotheses, for the part shown in Figure 5, which passed the pruning phase.

```
Primary_hypotheses = [hypothesis(path([4,5,6,7,8,9]),
                      view(ss2),
                      assignment([3,2,5,6,4,1])),

                      hypothesis(path([1,2,3,4,5]),
                      view(ts8),
                      assignment([1,3,4,5,2]))]
```

Notice how the hypotheses which survived pruning are indeed the ones which describe the primitives visible in the image of the part, namely the step-to-shoulder and the through slot.

4. Primitive Verification and Instantiation

We verify hypotheses by examining the geometric characteristics of the edges between surfaces, such as expected surface boundary shapes, and between surfaces and the background.

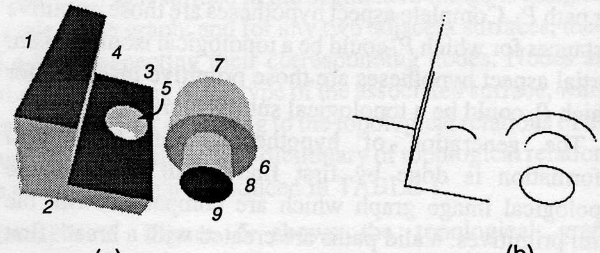
To examine an edge, we find those pixels which belong to the edge in question. If the set of edge pixels is non-empty and an edge was expected, we proceed to fit the pixels to a parametric equation of the expected geometric shape. We do not perform any edge tracing as part of the geometric verification process. Instead, fitting is done in the LMS error sense; if the fitting error is larger than a given threshold, then the verification fails and the hypothesis is rejected. The hypothesis is also rejected if we find pixels between two given surfaces—or between a surface and the background—even though the request indicated that no edge should exist.

If all geometric characteristics are satisfied, the primitive is declared verified. Otherwise, the primitive is rejected.

Instantiation could involve estimating the length of individual straight edges, collections of straight edges, or radii of cylinders. For straight edges, we estimate lengths directly from the extrema of the detected edge pixels. However, radii of cylinders are instantiated using the parameters of the cylindrical surfaces which are obtained during surface image segmentation.

Example 4. When geometric verification was applied to the part in Figure 5, the routines confirmed the existence of both primitive aspects. The instantiation routines found that the step-to-shoulder has dimensions x=0.139 m, y=0.146 m, and z=0.147 m, while the actual dimensions are x=0.15 m, y=0.15 m, and z=0.15 m. Similarly, the estimated dimensions for the through slot are x=0.296 m, y=0.146 m, and z=0.090 m, while the actual dimensions are x=0.3 m, y=0.15 m, and z=0.1 m.

Example 5. Another example of geometric verification and instantiation appears in Figure 6. In this case, all expected edges are detected and the primitives verified. The estimated length of the edge between surfaces 1 and 4 is 0.373 m; the actual length of that edge is 0.40 m. Similarly, the length of the edge between surfaces 1 and 2 is estimated as 0.098 m; the actual length is 0.10 m. Notice that a jump edge between cylindrical surface 5 and the background allows us to verify the through hole hypothesis. When the blind hole hypothesis is tested, the existence of an edge between surface 5 and the background forces the hypothesis to be rejected.



(a) (b)

Edge 4-1=0.373482 ***verified***
Edge 3-4=0.387613 ***verified***
Edge 2-1=0.098632 ***verified***
Edge 4-2=0.149284 ***verified***
Edge 3-2=0.199262 ***verified***
Edge 6-7 ***verified***
Edge 8-6 ***verified***
Edge 9-8 ***verified***
Jump edge 5=0.008495 0.041956***verified***
Jump edge 9=0.048145 0.071609***verified***

(c)

FIGURE 6. Verifying edges. (a) A segmented image; (b) edges traced on demand by the verification routines; and (c) a portion of the edge verification report.

5. Primitive Pose Estimation

Before indexing the object database, we need to estimate the pose of the verified primitives aspects. For this purpose, we consider each verified aspect as an independent object, and find its pose by comparing its surfaces with the surfaces of the respective model.

We adopt Flynn's [2] technique for translation and rotation estimation. For translation, the technique consists in solving a Least Mean Square problem, where each pair of corresponding image-model surfaces contribute one or more constraints. For rotation, the technique consists in obtaining a rotation estimate for each pair of corresponding surfaces, and calculating the average of the individual estimates.

Example 6. Following the pose estimation method, it was found that the local reference frame of the step-to-shoulder in Figure 5 has an orientation and translation *with respect to its model reference frame* given by

$$\boldsymbol{R}_1 = \begin{bmatrix} -0.573 & 0.818 & -0.002 \\ 0.409 & 0.287 & -0.866 \\ -0.708 & -0.498 & -0.499 \end{bmatrix}$$

$$t'_1 = \left[0.08\ 0.0\ 0.70\right].$$

This represents a variation of less than 5% with respect to its actual rotation and translation.

6. Work in Progress: Part Model Indexing and Pose Estimation

This section outlines the routines which we are currently developing, namely part model indexing and pose estimation.

Since part models are stored in the object database as *compositions* of instantiated primitives, the indexing of the object database requires two steps. First, a covering is created for the set of verified primitives. Each set in the covering is composed of primitives which are likely to be part of the same object because they appear to be physically connected. In the second step, these sets index the model database. A part model is indexed if for each class of primitive in the set, the total number of primitives of the same type in the model is at least the same; *and* the dimensions of a primitive in the set is approximately the same as those specified in the part model.

We invoke secondary, or even tertiary, topological hypotheses in cases where there is destructive topological occlusion or primitive interaction. If an indexed object cannot be fully explained with the verified primary hypotheses, we decide which primitives of the part model still remain unexplained. These primitives are looked up in the secondary topological hypotheses. The paths of the selected hypotheses are then tested for sub-isomorphism with respect to their primitive aspects. Those which are isomorphic are then verified and instantiated using similar procedures as those applied to primary topological hypotheses. Selection and testing of secondary hypotheses continues until one —or more, if needed— of them is found to be compatible with the part model. If the search ends in failure, we repeat the process with the tertiary hypotheses. If this search fails too, the part model is still considered a correct interpretation of the image, although we have not been able to fully explain it.

Finally, the part's pose is estimated by comparing the reference frames of the verified aspect primitives with the reference frames of the primitives included in the model.

7. Conclusions

We have presented a 3-D object recognition system for parts designed with a manufacturing primitive-based CAD. In this system, objects are modelled as organized compositions of instantiated primitives, while primitives are modeled at the topological and geometrical levels.

Primitive indexing is done in two steps: a fast, qualitative pattern matching, and a more attentive, relational graph matching.

This scheme offers several advantages. In terms of computational recognition time, the representation shifts the burden of sub-graph matching from the object model database to the CAD primitive model database. Since the size of the primitive database remains constant for a particular CAD system, and is considerably smaller than the object database, the matching costs are substantially reduced. Also, the addition of new items to the object database has little impact on recognition complexity both in terms of matching time and in modification of recognition heuristics. Finally, the interface of the recognition system with a CAD/CAM software is simplified.

8. References

[1] Arman, F., and J.K. Aggarwal, "CAD-Based Vision: Object Recognition in Cluttered Range Images Using Recognition Strategies," *CVGIP IU*, Vol. 58, No. 1, pp. 33-42, 1993.

[2] Flynn, P., and A.K. Jain, "CAD-Based Computer Vision: From CAD Models to Relational Graphs," *IEEE Trans. PAMI*, Vol. 13, No. 2, pp. 114-132, 1991.

[3] Mantyla, M., D. Nau, and J. Shah, "Challenges in Feature-Based Manufacturing Research," *Communications of the ACM*, Vol. 39, No. 2, pp. 77-85, 1996.

[4] Masotti, G., and T. Bombardi, "Automatic production of NC code for machining from features in generic parts," *Computing and Control Engineering Journal*, Vol. 3, No. 6, pp. 287-295, 1992.

[5] Marefat, M., M. Sandeep, and R.L. Kashyap, "Object-Oriented Intelligent Computer Integrated Design, Process Planning, and Inspection," *Computer*, Vol. 26, No. 3, pp. 54-65, 1993.

[6] Merat, F., and G. Radack, "Automatic Inspection Planning within a Feature-Based CAD System," J. Robotics and Computer Integrated Manufacturing, Vol. 9, No. 1, pp. 61-69, 1992.

[7] Villalobos, L., and F. Merat, "Neural Net Range Image Segmentation for Object Recognition," *Proc. Applications of Artificial Neural Networks IV*, SPIE 1965, Orlando, 1993.

[8] Villalobos, L., and F. Merat, "Recognition of Mechanical Form Features in Range Images," *CAISR Tech. Report 93-132, Case Western Reserve University*, 1993.

[9] Villalobos, L., and F. Merat, "3D Modeling and Indexing for CAD-based Object Recognition," *Proc. IEEE Int. Conf. Robotics and Automation*, San Diego, pp. 1965-1972, 1994

[10] Wang, W., and S. Iyengar, "Efficient Data Structures for Model Based 3-D Object Recognition and Localization from Range Images," IEEE PAMI, Vol. 14, No. 10, pp. 1035-1044, 1992.