Proceedings of the 1997 IEEE
International Conference on Robotics and Automation
Albuquerque, New Mexico - April 1997

Manufacturing Primitive-Based Object Identification Using Recognition-by-Components

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

It has been recognized that lead time and costs can be reduced by representing product designs in terms of manufacturing features such as surface finish, accuracy, or machining operations. Usually, a design environment supporting such representations is referred to as a primitive-based CAD system. This paper describes an object recognition system for range images which exploits the primitive-based CAD database information to reduce set up time and accelerate object identification. The heart of our work resides in modeling objects as collections of instantiated machining manufacturing primitives. This permits the implementation of a recognition-by-components approach using surface topology analysis. Thus, the main recognition task is broken down into the less stringent ones of primitive identification and instantiation, followed by the actual object model indexing. The result is faster recognition as well as automatic generation of object models from the CAD's output databases.

1. Introduction

It is usually agreed that over 70% of a new product's total cost has been committed by the end of the design stage. It has also been recognized that lead time and costs can be reduced, while preserving design intent, by representing product designs in terms of manufacturing features [4, 8]. These features could describe, for example, the required fabrication materials, surface finish, dimension accuracy, and/or machining operations [5, 6]. In general, a Computer Aided Design (CAD) system supporting such design representations will be referred to as a *primitive-based CAD system* [7].

From a computer vision point of view, the availability of CAD data also gives the opportunity to tackle two serious problems in the application of vision to manufacturing automation, namely the lengthy preparation of object recognition models, and the difficult maintenance and modification of model databases. Several researchers report that vision systems for assembly and inspection can be constructed faster and maintained more efficiently by exploiting the geometry-rich CAD models of the objects [1, 2, 9].

This paper describes an ongoing research effort aimed at developing a 3-D object recognition system for parts which have been (or could be) designed with a primitive-based CAD. In particular, we exploit the geometric information contained in the CAD's design, which is captured by the manufacturing (machining processes) primitives. The heart of the approach resides in modeling objects as collections of instantiated manufacturing primitives. This allows the application of a recognition-by-components method in which the main recognition task is broken down into the less stringent ones of primitive identification, instantiation, and pose estimation, followed by the actual object model indexing. The result is not only a faster recognition but also the *automatic* generation of objects models by analyzing the CAD's output database.

In section 2, we present the primitives currently supported by the recognition system, and describe our modeling schemes for both primitives and objects. Primitive hypotheses generation is described in Section 3, while verification and geometric instantiation are explained in Section 4. Primitive pose estimation is presented in Section 5, with object models retrieval in Section 6. Finally, conclusions are given in Section 7.

2. Model Construction

This section describes our representation schemes for both the primitives and the parts constructed with them. In our work, objects are modeled as a collection of instantiated manufacturing primitives, and these primitives are, in turn, modeled following a viewer-centered approach. This type of hierarchical modeling resembles the standard design procedure followed in a primitive-based CAD system [6, 7]. Thus, it facilitates the automatic construction of models for recognition through a straight forward analysis of the CAD's output databases.

Primitive representation is examined in the context of the nine manufacturing primitives shown in Figure 1.

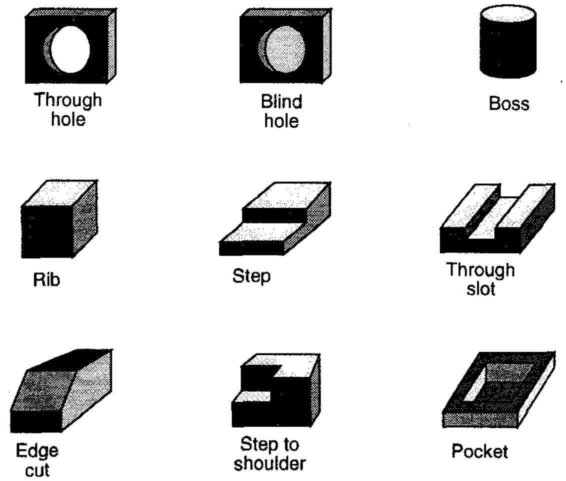


FIGURE 1. Supported manufacturing primitives.

2.1 Primitive Topological Modeling

We model the primitives following a *topological* viewer-centered approach. In our approach, the surface topology of each aspect is represented by an attributed graph which includes 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 [11]. 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 or views which we have numbered for identification purposes. For example, a blind hole has three aspects, the first one of which (view 1) is depicted in Figure 2 together with its 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.

2.2 Qualitative Aspect Composition Database

We store the topological models in two databases: one with the topological graphs themselves, and another with *patterns* which *qualitatively* describe the topology of complete and partial aspects. The latter is referred to as the qualitative aspect composition database (QACD). Its construction is as follows.

Let the set of surfaces of a particular aspect be $M=\{M_k; k=1, 2, ..., K\}$, where K is the number of surfaces in the aspect, and let the set of non-empty subsets of M be $M'=\{M'_j; j=1, 2, ..., J; M'_j \subseteq M; |M'_j| \neq 0\}$. We refer to a layered organization of M' as a *cumulative hierarchy* of M. Every layer in the hierarchy includes those elements of M' with equal cardinality. Thus, at the top of the hierarchy, we have the aspect itself; next, we have the elements of M' with cardinality K-1; and so on.

We create a pattern to qualitatively describe each subset M'_j. The pattern includes two parts, one to represent the types of surfaces in M'_j, and the other to represent their topological connectivity. This representation only includes the sum of each type of surface and each type of connectivity. A pattern has also attached the label of its parent aspect as either a complete hypothesis —for the pattern of the aspect itself, or as a partial hypothesis —for all remaining patterns.

Sometimes the same pattern can appear in the cumulative hierarchies of different primitives' aspects. Because of this, once all hierarchies have been created, we proceed to group together those patterns which are identical, and to combine their complete and partial hypotheses in two independent lists: the *complete aspect hypotheses* list, and the *partial aspect hypotheses* list, respectively. Hence, if `plane` refers to the total number of planar surfaces, `cylcx` to the total number of cylindrical convex surfaces, and so on, then a pattern in the database can be represented by the PROLOG clause

```
pattern(surfaces((plane,cylcx,cylcc)),
        connectivity((lec,lic,lid,
                      pec,pic,pid,ang,pse)),
        class([hc],[hp]])).
```

where hc and hp are the lists of hypotheses for complete and partial aspects, respectively.

Example 1. Let us consider the qualitative pattern description for aspect 1 for a hole, whose cumulative hierarchy is depicted in Figure 3. Here we analyze three representative patterns of the seven total.

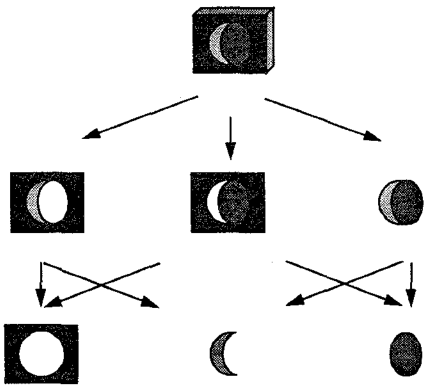


FIGURE 3. Decomposition of a primitive aspect. (a) Aspect 1 of a blind hole; (b) partial aspects at the first decomposition level; and (c) partial aspects at the second decomposition level.

The complete aspect is at the top of the hierarchy. Since it is not a partial component of any other aspect, its pattern has associated only the complete aspect hypothesis for blind hole (view 1).

Next, we have the partial views for sets of two surfaces. *Topologically*, the first partial view is identical to both the first aspect of a through hole and the second aspect of a blind hole. Hence, the pattern will have complete aspect hypotheses for through hole (view 1) and blind hole (view 2), and a partial aspect hypotheses for blind hole (view 1). Notice how a complete aspect can be a partial hypotheses for another aspect.

Individual surfaces appear at the bottom of the hierarchy. The concave surface is a component of the first aspect of a through hole, and the first aspect of a blind hole. Consequently, its pattern will have partial hypotheses for through hole (view 1), and blind hole (views 1 and 2).

Summarizing, the portion of the database which includes the three patterns looks as follows

```
                             ...
pattern(surfaces((2,0,1)),
        connectivity((1,1,1,0,0,0,0,0)),
        class([[bh1],[]])).

pattern(surfaces((1,0,1)),
        connectivity((1,0,0,0,0,0,0,0)),
        class([[th1,bh2],[bh1]])).

pattern(surfaces((0,0,1)),
        connectivity((0,0,0,0,0,0,0,0)),
        class([[],[th1,bh1,bh2]])).
```

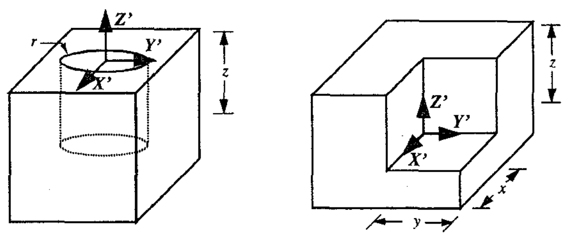
2.3 Primitive Geometric Modeling

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 characteristics of their pertinent edges (jump, straight, circular arcs, etc.)

2.4 Part Modeling

Each part is described as an organized aggregate of instantiated generic 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.

Each primitive has a convention to instantiate its characteristic parameters, as Figure 4 illustrates for a blind hole and a step-to-shoulder.

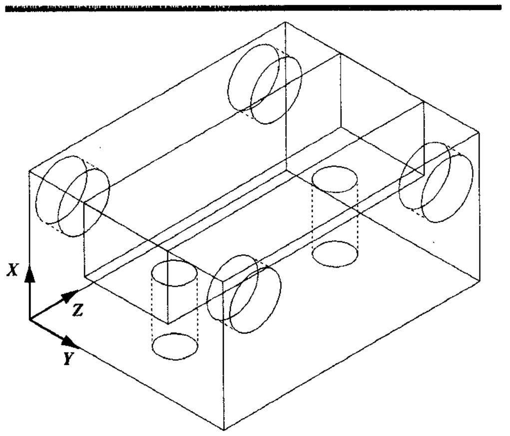


(a) (b)

FIGURE 4. Primitive geometric parameters and definition of local reference frames. (a) A blind hole is completely characterized by its radius, depth, and axis orientation. (b) The geometry of a step-to-shoulder is defined by the dimensions of the cut.

A primitive's pose is instantiated by first selecting a global reference frame that has 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). The model of a pulley bracket which was automatically generated from the output of a manufacturing primitive-based CAD system [7] appears in Figure 5.



(a)

```
model(part_id(2,pulley-bracket),
      stock(2.00,3.00,4.00),
      primitives([th_hole,bl_hole,th_slot]),
      description([
            th_hole(2,[(1,param([0.25]),
                        pose([0,0,-1],[0,1,0],[1,0,0],[0,1.5,3.25])),
                     (2,param([0.25]),
                        pose([0,0,-1],[0,1,0],[1,0,0],[0,1.5,0.75]))]),
            bl_hole(4,[(1,param([0.4,0.35]),
                        pose([0,1,0],[0,0,1],[1,0,0],[1.5,2.65,0.5])),
                     (2,param([0.4,0.35]),
                        pose([1,0,0],[0,0,-1],[0,1,0],[1.5,0.35,3.5])),
                     (3,param([0.4,0.35]),
                        pose([1,0,0],[0,0,-1],[0,1,0],[1.5,0.35,0.5])),
                     (4,param([0.4,0.35]),
                        pose([0,1,0],[0,0,1],[1,0,0],[1.5,2.65,3.5]))]),
            th_slot(1,[(1,param([1.3,1]),
                        pose([0,0,1],[0,1,0],[-1,0.0],[1,0.85,4]))])])).
```

(b)

FIGURE 5. A part and its model. (a) Isometric view of a pulley bracket with its global reference frame, as displayed by the CAD system; (b) automatically generated part model.

The advantage of this modeling style is that the part database becomes a collection of compact, simple linguistic descriptions. Furthermore, the addition of new models to the part database does not involve any significant modification 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.

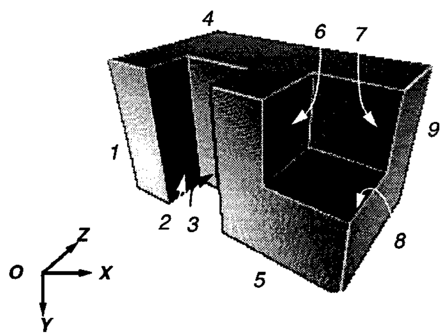
3. Topological Primitive Indexing

On line, our recognition strategy starts by recognizing and instantiating the generic primitives, and then using this information as indexing keys to the objects' model database. The advantage of this approach resides in the reduction of search time due to the smaller size of the primitives database with respect to the object database.

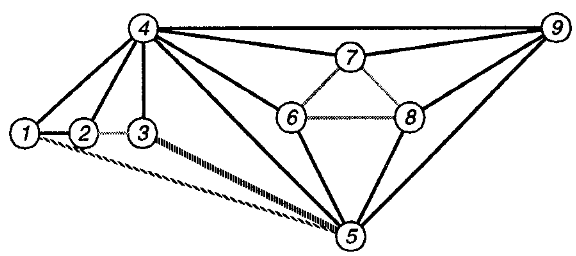
3.1 Topological Graphs

Recognition starts with a surface-based segmentation. An undirected, attributed graph G is then used to describe the topological relations between the surfaces of the segmented image [10]. Each node in the graph corresponds to a surface in the image, and is labeled according to the surface's type; similarly, links are labeled according to the topological relations between surfaces.

Example 2. Figure 6 shows the topological graph constructed from the segmented image of a simple part.



(a)



(b)

FIGURE 6. Topological graph of a segmented image. (a) Image with individual surfaces identified; (b) its topological graph.

3.2 Preliminary Topological Indexing

Topological hypotheses are generated by first finding all subgraphs P_j in G which are compatible with one or more aspects. A subgraph P_j is considered to be compatible with an aspect if there is supporting evidence that its surfaces *could* match the topology of the aspect.

Matching the topology of each possible subgraph of G

later by matching the topological graph of the indexed aspect against the topology of the surfaces in the path. Notice that the correct hypotheses for the step-to-shoulder (view 2) and the through slot (view 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 desired solution to our topological isomorphism problem requires the satisfaction of three types of constraints: *uniqueness correspondence*, *shape correspondence*, and *topological correspondence* [11].

Example 5. The following is the list of primary topological hypotheses, for the part shown in Figure 6, 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 the ones which describe the primitives visible in the image of the part, namely the step-to-shoulder and the through slot. The `assignment` portion indicates surface correspondence between the path and the model of the aspect. For example, surface 4 in the image corresponds to surface 3 in the model of the aspect 2 for a step-to-shoulder.

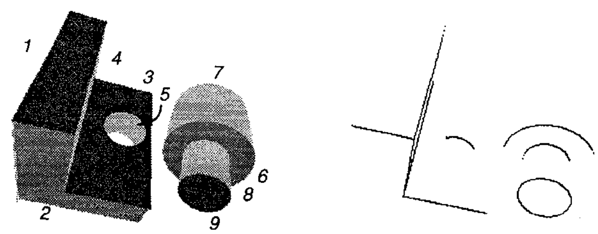
4. Primitive Verification and Instantiation

We verify hypotheses by comparing the geometric model of the indexed aspects against the geometry of the edges identified in the image. In this setting, verification becomes a model-driven task. We do not perform any edge tracing as part of the verification process. Instead, fitting is done in the LMS error sense. If all geometric characteristics are satisfied, the primitive is declared verified. Otherwise, the primitive is rejected.

Instantiation can involve estimating the length of individual straight edges, collections of straight edges, or radii of cylinders.

Example 6. The geometric verification routines confirmed the presence of both primitive aspects in Figure 6. 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 7. Another example appears in Figure 8, where 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 8. 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

We estimate the pose of every verified primitive aspect by considering it as an independent object, and finding the pose of its local reference frame with respect to the image global reference frame. This is done by comparing the direction vectors of corresponding image-model surfaces, following Flynn's [2] techniques to estimate translation and rotation.

Example 8. It was found the local reference frame of the step-to-shoulder in Figure 6 has the orientation and translation *with respect to its model reference frame* against the aspects' topological graphs would be a computationally expensive task. Instead, we find the subgraphs P_j with a breath first search on the *connected paths* of each node in the image graph. Suppose a connected path in G corresponds to a subgraph P_j. We decide whether P_j could be compatible with one or more aspects by first extracting the qualitative pattern for its surfaces, and then matching this pattern against the contents of the QACD. This way, a connected path of the image graph can quickly be *qualitatively* matched against the aspects. Notice how we can then associate two sets of hypotheses to each path P_j: the complete aspect hypotheses list (H_{cj}), and the partial aspect hypotheses list (H_{pj}). The search in each branch of the tree is pruned whenever the respective qualitative pattern does not match any pattern in the QACD.

Example 3. Consider the topological image graph shown in Figure 7. The process of topological hypotheses generation starts by finding connected paths for node 1. Since node 1 represents a cylindrical concave surface, a search in the QACD reveals that its qualitative pattern could partially match the graphs for a through hole (view 1), and blind hole (views 1 or 2). See Example 1.

Next, we add an adjacent node, in this case node 2, and form the connected path [1,2]. By comparing the qualitative pattern of [1,2] against the QACD, we find that it too could be explained as being a portion of view 1 for a blind hole, or the complete version for a through hole (view 1), or a complete version for a blind hole (view 2).

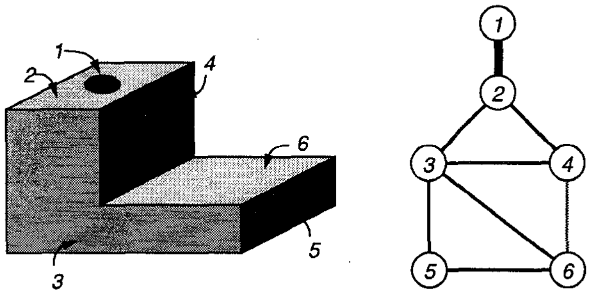
We continue expanding and add node 3, to form the connected path [1,2,3]. Now the comparison reveals that [1,2,3] cannot be explained in terms of the existing aspects; hence, [1,2,3] is discarded. The same happens if we expand [1,2] with node 4, and form the subset [1,2,4]. At this point, we cannot create any other connected path which includes node 1, and stop finding paths for that node. Consequently, from these analysis we can so far post the following preliminary hypotheses:

- surfaces [1,2] correspond to a complete version of a blind hole (view 2), or a through hole (view 1)
- surfaces [1,2] correspond to a partial version of a blind hole (view 1)
- surface [1] corresponds to a partial version of a through hole (view 1), or a blind hole (views 1 or 2).

Other nodes are analyzed, and hypotheses posted, in a similar fashion.

3.3 Topological Hypotheses Organization

As a result of their qualitative selection, not all the preliminary hypotheses have the same image description potential. For instance, in the previous example we would



(a) (b)

FIGURE 7. Primitive indexing. (a) A surface segmented image of an object, and (b) a graph representing the surfaces topological attributes.

intuitively consider path [1,2] to have more explanatory potential than path [1], simply because it includes more surfaces.

Based on heuristics explained elsewhere [11], 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 in case of substantial destructive occlusion.

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

Example 4. The following is the list of primary topological hypotheses generated for the part depicted in Figure 6.

```
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])]
```

Since they are generated through a qualitative matching, some of these hypotheses are incorrect; they will be pruned

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

$$\boldsymbol{t}'_1 = \begin{bmatrix} 0.08 & 0.0 & 0.70 \end{bmatrix} .$$

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

6. Object Model Indexing

Indexing of the object model database requires two steps. First, the verified primitives are heuristically grouped into sets of primitives which are or could be physically interconneced.We refer to these sets as the *object primitive sets*. In the second step, the sets are treated as keys to index the object database.

The object primitive sets are generated based on a physical connectivity heuristic. Due to our definitions, two surfaces are physically connected if there is a link between them in the topological image graph which is a member of the list of continuous links $L=$ `{lic, lec, pic, pec, or ang}`.

Example 9. Only one object primitive set was formed with the verified primitives from Example 6; naturally, it included both the step-to-shoulder and the through slot.

Example 10. Correctly, two object primitive sets were formed with the verified primitives shown in Figure 8. The first one included the step and the through hole, while the second one included the two bosses.

Every object primitive set indexes the object database, and extracts models which could satisfy its composition. A model is indexed as a hypotheses if:

- for each type of primitive in the object primitive set, the total number of primitives of the same type in the part model is at least the same; *and*
- the dimensions of a primitive in the object primitive set is approximately the same than those specified in the part model. We currently tolerate a 10% difference between the primitive's dimensions and the model's.

As part of our future work, we will use the sets of instantiated primitives to verify objects' hypotheses as well as estimate the objects' pose.

7. Concluding Remarks

We have described a recognition system for objects whose geometry can be represented in terms of 3-D mechanical manufacturing primitives. In our approach, objects are modeled as compositions of instantiated manufacturing primitives, and these primitives are in turn modeled with topological viewer-centered techniques.

Recognition is broken down into two main phases. The first one consists of identifying, verifying and instantiating the manufacturing primitives visible in the image, while the second one consists in using the verified primitives to index the object model database, and estimate the pose of the recognized objects.

Our technique not only accelerates recognition, but also facilitates the addition of models into the object database without modifying the general recognition methods or significantly increasing computational time. Furthermore, the object models can be automatically generated by analyzing the CAD design files.

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] Kak, A.C., A.J. Vayda, R.L. Cromwell, W.Y. Kim, and C.H. Chen, "Knowledge-based robotics," *Int. Journal of Production Research*, Vol. 26, No. 5, pp. 707-734, 1988.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] Requicha, A., "Geometric Reasoning for Intelligent Manufacturing," *Communications of the ACM*, Vol. 39, No. 2, pp. 71-76, 1996.

[9] Sobh, T., J. Owen, C. Jaynes, M. Dekhil, and T. Henderson, "Industrial Inspection and Reverse Engineering," *Computer Vision and Image Understanding*, Vol. 61, No. 3, pp. 468-474, 1995.

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

[11] 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