3DPO's Strategy for Matching Three-Dimensional Objects in Range Data

Patrice Horaud and Robert C. Bolles

SRI International
Menlo Park, California USA

ABSTRACT

A strategy for recognizing and locating three-dimensional objects in range data is presented. The strategy combines information derived from models of the objects and edges and surfaces detected in the data to efficiently match objects. Given a set of objects to be found, the set of object features are partitioned into subsets having similar intrinsic properties. An ordered tree of features to be considered is set up for each subset. These search trees are designed to maximize the use of the information as it is obtained and minimize the time required to recognize objects. A detailed example of this approach being used to recognize moderately complex castings in a jumble is presented.

INTRODUCTION

How does a person find a can opener in a drawer of kitchen utensils? If it is partially occluded by other objects, as it probably is, they recognize one of its "local" features, such as a slightly curled, pointed piece of sheet metal, and then use it to hypothesize the position and orientation of the whole object. The more distinctive the visible features are, the faster a person can find it.

We are interested in tasks like these not only because we want to explore how people perform them, but also because we want to develop techniques for performing similar industrial tasks. In this paper we are particularly concerned with the problem of recognizing and locating identical objects jumbled together in a bin (see Figure 1). We use models of the objects to find them in range data. By working on this class of difficult tasks we expect to develop general-purpose techniques for recognizing and locating partially visible objects.

For industrial tasks one of the most important merits of a technique is its speed. One way to achieve speed is to locate as few object features as possible and extract as much information from each one as possible. For example, after finding a feature, its identity can be used to suggest the next feature to be located and its position can constrain the region to be searched. A typical strategy of this type is:

Locate a distinctive feature of the object to be found.

Use that feature's position to suggest where to look for a second feature to verify the first.

Use the two features to predict a third feature that completely constrains the position and orientation of the object.

Since some of the predicted features may not be visible or the feature detectors may miss some visible ones, alternatives have to be provided. Therefore, a complete strategy is an ordered tree of features and associated with each feature are two sets of deductions, one to be made if the feature is found, and the other if it is not. The process of recognizing and locating an object is thus a tree search.

As already mentioned, one way to reduce the amount of search is to use the positions of located features to reduce the regions to be searched for additional features. Another way is to use the topology of the features detected in the data to provide candidates to be considered. For example, in range data, if one edge of a surface has been found and the next feature to be found is a parallel edge on the same surface, it is signifi-cantly faster to examine the other edges of that surface than to examine all edges located in the data.

As with all tree searches it is important to order the alternatives according to their expected utility. To minimize execution time it is possible to order the alternatives in advance. One of our goals for the 3DPO system is to develop such an off-line planning system. However, in this paper we concentrate on the development of strategies that capitalize on the information contained in object models and derived from detected features.

Previous approaches for locating partially visible objects in range data have generally not been designed to use the full range of available information. They have usually concentrated on one type of feature, such as surface patches (e.g., see [1], [2], and [3]), edges (e.g., see [4]), or simple shapes (e.g., see [5], [6], and [7]). Recently, Dave Smith at CMU has developed a

CH2008-1/84/0000/0078$01.00©1984 IEEE

system that builds descriptions of such objects as pans and shovels from a combination of surface and edge features [unpublished].

In this paper we describe a "grow-a-match" search strategy for recognizing objects in range data that tries to maximize the use of the information available from both object models and data. We describe the metric and topological information associated with planar and cylindrical features, and then show how to use it to construct strategies for locating the castings shown in Figure 1 efficiently.

APPROACH

The task is to recognize and locate objects jumbled together in a pile. The approach, as outlined above, is to locate a key feature, and then add one feature at a time until the object can be reliably and precisely located. In this grow-a-match approach there are four contributions a new feature can make:

- Reduce the number of interpretations for a cluster of located features.
- Verify an interpretation.
- Determine some unknown degrees of freedom associated with an interpretation.
- Increase the precision with which the degrees of freedom can be computed.

A single feature can make more than one contribution. For example, a new feature could verify an interpretation and increase the precision.

An unconstrained object in a jumble has 6 degrees of freedom associated with its position and orientation, 3 displacements and 3 rotations. Different types of object features constrain different degrees of freedom. For example, the circular edge at the end of right circular cylinder protruding from an object determines 5 of the object's 6 degrees of freedom. The only unknown is the rotation about the axis of the cylinder. Locating a straight edge on an object and the two planes on either side of it determines 4 degrees of freedom. Thus, circular edges provide somewhat more information than straight edges.

For objects with circular and straight edges, circular edges are generally preferred over straight edges for two additional reasons. First, the radius of a detected circle can be used to reduce the set of possible object circles it could match. Straight edges do not have similar properties. The size of the dihedral angle at an edge may be helpful, but circular and straight edges share this property. Second, since circles have an inside and an outside, the relative position of the cylindrical surface with respect to the planar surface can be used to further reduce the number of interpretations. It should be pointed out, however, that for a specific object a straight edge may still be preferred over a circular one for some reason such as the straight edge is the only feature that can determine one of the object's degrees of freedom.

The ordering of features is a function of several factors including their expected contributions, the cost of detecting them, and the likelihood they will be detected. Some features are inherently easier to find. If two features provide essentially the same contribution, but one is easier to find, it should be ranked ahead of the other one. If a feature is often missed by the feature detectors, its rank should be lower than for one that is consistently found. Another factor to be considered is the predicted visibility of a feature. For example, if one or more features have been found, some features may be eliminated from the list of features to be considered because they are on the side of the object away from the sensor. Faugeras et al. have used this type of reasoning to reduce their tree searches [3].

Given a feature to be found, how should the recognition system generate candidates and test them? If the feature is the first one to be found and the recognition system is running on a sequential machine, the system is forced to examine each feature sequentially. If a feature's intrinsic properties match those of the desired feature, it is accepted. Some features have more intrinsic properties than others. For example, the end of a right circular cylinder has several properties that can be checked. A matching edge must lie in a plane and fit a circle with approximately the expected radius. If the adjacent surfaces are visible, their sizes and relative positions must match those of the desired feature.

If the recognition system is looking for a feature after having already found some other features, the located features can be used to reduce the search. This can be done either by restricting the portion of the data to be analyzed or by specifying the topology of the new feature with respect to the old ones. The located features can also specify criteria that the candidates must meet to be a match. For example, they can specify a range of relative orientations and a range of distances with respect to each of the located features. In the next section we show how this type of constraint can be used to form efficient recognition strategies.

EXAMPLE

The example task is to use the model shown in Figure 2 to locate castings in a jumble such as the one in Figure 3. The data to be used is the range data in Figure 4. It is a height image in which the height above the table is encoded so that higher points are brighter. This height image and the intensity image in Figure 3 were gathered simultaneously by a White Scanner 100A, which uses triangulation to compute the three-dimensional coordinates of points on a projected plane of light. Each row in these images corresponds to one position of the light plane as it is scanned across the scene. Thus, the images are views of the scene as seen from the light source.

The model in Figure 2 is constructed of planar and cylindrical components. Planes and cylinders were used because they are common components of machined and cast objects and they can be modeled easily mathematically. The model contains 7 full cylinders, 8 partial cylinders, and 25 planar patches, all of which are bounded by 32 circular arcs and 28 straight lines. These numbers are large enough to preclude the straightforward matching strategy that compares each observed feature (i.e., line, arc, plane, or cylinder) to each model feature and tries to find the largest set of consistent matches. The combinatorial explosion inherent in this search, however, can be reduced dramatically by measuring properties of the observed features, such as the ones described in the last section, and restricting the matches to those between features with similar properties.

Object recognition is a two-part process: a low-level data-driven analysis followed by a high-level model-directed tree search. The low-level analysis locates edges in range data, partitions them into circular arcs and straight lines, and then characterizes the surfaces adjacent to each arc and line segment. The high-level process selects one after another of these edge features and tries to use them to recognize objects. For each "focus feature" it produces a list of model features that match the feature's properties. Then it tries to narrow down this list by growing a cluster of recognized features about the focus feature. To maximize the use of the information as it is obtained, the program tries to add one feature at a time. If it is difficult to select a single feature to add, the strategy considers a small set of features and uses a maximal clique algorithm to find the largest subset of these features that are mutually compatible. Since the time to compute maximal cliques grows exponentially with the number of features, it is important to minimize the number of features [8].

We begin our low-level analysis of the range data by locating edges because they contain more information than surface patches, and we are able to locate them more reliably. To locate edges we detect discontinuites in the range data and link them together. We classify each discontinuity as a convex edge, a concave edge, a jump, or a shadow. Shadows are eliminated from further processing because they are assumed to be artifacts of the camera and lighting geometries. Figure 5 shows the non-shadow edges located in Figure 4.

The edges are partitioned and classified in a multistep process. First, they are partitioned into subedges lying in planes. Second, the subedges are partitioned into straight segments and circular segments. (Figure 6 shows the straight lines and circular arcs detected along the edges in Figure 5.) Third, the visible surfaces on either side of the segments are analyzed. The analysis classifies each surface according to its type and measures a few simple properties such as the maximum excursion of the surface from the edge. A circular edge is expected to have one planar surface and one cylindrical surface adjacent to it. Straight segments may be the intersection of two planes, the intersection a plane and a cylinder, or the tangential edge of a cylinder. (Figure 7 shows some of the cylinders located from tangential edges.) After the surfaces have been classified, they are refitted with as much data as possible and the updated surface equations are used to improve the estimates of the parameters of the lines and circles. This completes the low-level processing.

For the high-level processing the set of features associated with the objects to be recognized are partitioned into subsets having similar intrinsic properties. For example, if an object has two half-inch holes and four eighth-inch holes, the half-inch holes are grouped together into one subset and the eighth-inch holes are placed in another. Each subset has its own strategy for locating an object by growing a match from a feature that is similar to the ones in the subset. Thus, if a half-inch hole is found, one strategy is used; if an eighth-inch hole is found, a different strategy is used. The strategies depend on many factors including the number of object features in the subset and the degrees of freedom constrained by the features.

As noted in the last section, circular edges are generally more useful than straight edges because they have more properties to narrow down their list of matches. This is true for the casting in Figure 2 even though it has more circular edges than straight edges. The circles on the casting are preferred over tangential edges as well because they are easier to detect.

Figure 8 lists 15 of the circular arcs on the casting and their intrinsic properties. The other 17 arcs are not included because they are either too small to detect or are not visible. The horizontal lines in the table delimit the sets of features that have similar properties. Notice, for example, that there are four circles with radii of approximately 2 inches, but they are partitioned into two subsets because two of them have a planar surface on the inside of the circle and two of them have it on the outside.

Figure 9 shows an arc with a radius of approximately 2 inches that was detected by the low-level system. It has a planar surface on the outside, which means it must be either part of the SHELF-BOTTOM or part of the INSIDE-BASE. Since its length is significantly longer than 1.7 inches the system can eliminate one of these interpretations. To verify its hypothesis, the system tries to locate concentric circles and finds the ones shown in Figure 10. These circles verify the hypothesis. The system then proceeds to use the initial circle to compute 5 of the casting's 6 degrees of freedom. To determine the sixth degree of freedom it has to locate additional features. Any feature that is not symmetric about the axis of the cylinder would be sufficient. Unfortunately, the best features are the line segments, which are all quite similar. Therefore, instead of selecting one feature, the strategy is to locate all lines in that portion of the data and apply the maximal clique algorithm to find the largest set of compatible matches. Figure 11 shows the set of line segments that are in the same plane as the circle and are approximately the right distance from the center of the circle. There are five lines, each with two or three interpretations, which implies a graph containing 12 nodes to be analyzed by the maximal clique algorithm. Figure 12 shows the final match, which is based on the initial circle and four of the line segments.

In Figure 13 another circular arc is shown with its set of line segments for deteriming the rotation. In this case the maximal clique algorithm finds three line segments mutually compatible with the circle. Figure 14 shows the final hypothesis.

If all the circles have been analyzed and some data is left unexplained, the system starts considering the second best focus feature, which is a tangential edge of a cylinder. One of these edges determines 4 degrees of freedom. The basic recognition strategy for them is to locate other cylinders and circles that have axes almost collinear with the initial cylinder. Locating one additional feature is generally enough to compute 5 degrees of freedom, and then the strategy is the same as for circles: locate a feature to determine the sixth degree of freedom. Figure 7 shows the cylinders found from tangential edges. Figure 15 shows a cylinder and a compatible circle. Figure 16 shows a hypothesis based on the 5 degrees of freedom computed from the cylinder and circle. Finally, Figure 17 shows hypotheses for 6 of the 7 visible castings. The system missed the seventh one because it only found one of its features.

DISCUSSION

The edge-based strategies described in this paper are made possible by the fact that we can reliably locate edges and measure properties of the adjacent surfaces. However, since our data is gathered by a triangulation system, it rarely contains more than two surfaces that are directly connected. Usually a "missing data" area intervenes between surfaces. Therefore, any attempt to grow matches topologically past a couple of surfaces is likely to fail.

In the past most vision researchers have avoided objects with several features because they wanted to keep the images as simple as possible. We have found, on the other hand, that an abundance of features is generally helpful because it means that there is almost always a nearby feature to help identify one with many interpretations. If all the objects are parallelepipeds, all the features look the same. The nearby features are not much help either because they all look the same too. However, if an object has 30 different features, then finding one is often enough to make a hypothesis and finding two is generally enough to make a hypothesis and verify it. An indication that the features on the casting provide significant information is that the average depth of a strategy tree is about three. A typical strategy includes a focus feature, a feature to verify the hypothesis, and a third feature to determine the last degree of freedom or two.

The most time-consuming part of the recognition process is the analysis of the surfaces adjacent to the edges. It is important to determine the surface types, but less important to produce exact fits. Rough estimates of the surface parameters are generally sufficient. The objects are located precisely by combining features that are some distance apart as opposed to relying on one feature's parameters. Therefore, it appears that it would be more efficient to develop some techniques that quickly determine surface types than to spend the time to perform costly iterative fitting.

It is interesting to note that the problem of locating objects with six degrees of freedom in range data is remarkably similar to the problem of using grayscale images to locate objects constrained to lie on a plane parallel to the image plane. In both cases one feature determines most of the degrees of freedom. For example, a circular hole in a grayscale image determines two of the three degrees of freedom. A circular edge in range data determines five of the six degrees of freedom. Another similarity is that a partial topology of features can be determined. In range data the edge-surface-edge connectivity provides a direct way to grow clusters of related features. In grayscale images the corner-line-corner connectivity along edges provides a similar capability.

A key to the success of a recognition strategy such as the one described in this paper is the preplanning that enumerates the features and forms the strategy trees for focus features. At present the enumeration is done automatically from the model, but the formation of strategies and the ranking of focus features is done interactively. In fact, the strategies are currently represented as special-purpose programs instead of data structures that are interpreted. We are currently designing an AND/OR-type tree structure to represent strategies, and we are exploring automatic ranking techniques for ordering the features.

Industrial systems, to be practical, have to be robust. For the 3DPO system this means that the high-level system has to work in spite of the low-level system missing features and finding extra ones. It recovers from most missing features by focusing on other features of the object. If several features on an object are missed or are obscured by other objects, the system may take a while to recognize the object because it has to cycle through its list of features. If all the major features are missed, the system will miss the object.

Extra features slow the analysis down a little because they force the system to explore hypotheses that it cannot verify. However, we have not found this to be a significant problem. In fact, we usually set the thresholds for accepting features relatively low and rely on the high-level system to filter out the extra ones. Low thresholds are beneficial because they let in features that otherwise would have been missed and the false features are quickly thrown out because they do not form clusters of features that are consistent with the objects.

The tests we use for checking interpretations for pairs of features are extensions of the two-dimensional point-to-point tests used in the local-feature-focus system [9] and the three-dimensional point-to-plane tests used by Grimson and Lozano-Perez [10]. Since the observed features are segments of lines, circles, and cylinders, the tests are segment-to-segment matches in the sense that they can use the lengths of the features to constrain the amount of sliding one feature can do along a matching feature. Long features constrain the sliding more than short ones. So far we have not tried to develop a minimum set of tests. We have simply implemented a set of inexpensive tests to eliminate obvious mismatches.

In the future we plan to increase the set of features to include such things as ellipses and implement models of more objects. We are particularly interested in developing a larger set of techniques for using information derivable from features. We believe that a general-purpose system will need to be able to use a wide range of information and make a variety of different types of deductions. We do not believe that a single technique will be general enough to recognize a large class of objects efficiently.

We also plan to investigate ways to perform configuration understanding, which means developing a model of a scene in terms of the detected objects. Such a model would make it possible to decide which object is on top of the jumble or which objects will move if one is picked up.

REFERENCES

1. Duda, R.O., D. Nitzan, and P. Barrett, "Use of Range and Reflectance Data to Find Planar Surface Regions," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-1, No. 3, pp. 259-271 (July 1979).
2. Oshima, M., and Y. Shirai, "Object Recognition Using Three-dimensional Information," Proc. Seventh IJCAI, pp. 601-609, Vancouver, B.C. (August 1981).
3. Faugeras, O. D., and M. Hebert, "A 3-D Recognition and Positioning Algorithm Using Geometrical Matching Between Primitive Surfaces," Proc. Eighth IJCAI, pp. 996-1002, Karlsruhe, Germany (August 1983).
4. Nevatia, R., and T.O. Binford, "Description and Recognition of Curved Objects," Artificial Intelligence, Vol. 8, pp. 77-98 (1977).
5. Popplestone, R.J., et al., "Forming Models of Plane-and-Cylinder Faceted Bodies from Light Stripes," Proc. Fourth IJCAI, pp. 664-668, Tbilisi, Georgia, USSR (September 1975).
6. Shirai, Y., and M. Suwa, "Recognition of Polyhedrons with a Range Finder," Proc. Second IJCAI, pp. 80-87, London, England (August 1971).
7. Nitzan, D., A.E. Brain, and R.O. Duda, "The Measurement and Use of Registered Reflectance and Range Data in Scene Analysis," Proc. IEEE, Vol. 65, pp. 206-220 (1977).
8. Bolles, R.C., "Robust Feature Matching through Maximal Cliques," Proc. SPIE Technical Symposium on Imaging Applied to Automated Industrial Inspection and Assembly, Washington, D.C. (1979).
9. Bolles, R.C., and R.A. Cain, "Recognizing and Locating Partially Visible Objects, The Local-feature-focus Method," International Journal of Robotics Research, Vol. 1, No. 3, pp. 57-82 (1982).
10. Grimson, W.E.L., and T. Lozano-Perez, "Model-Based Recognition and Localization from Sparse Range or Tactile Data," to appear in International Journal of Robotics Research (1984).

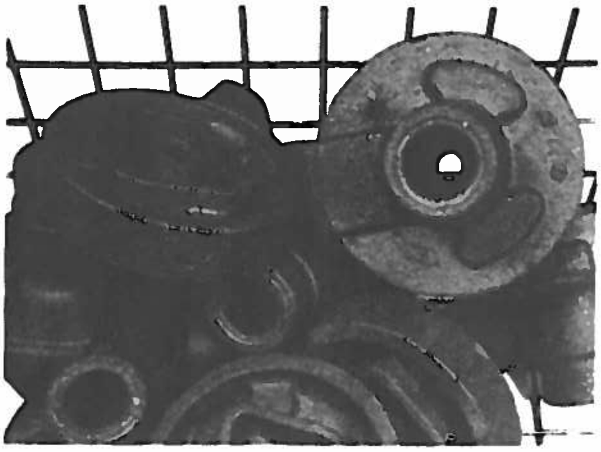

FIGURE 1 BIN OF CASTINGS

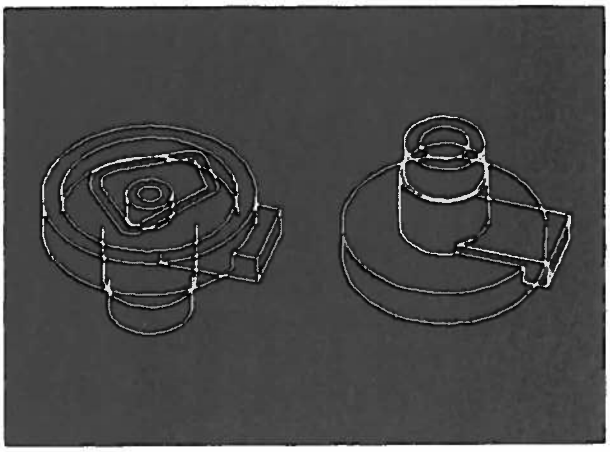

FIGURE 2 MODEL OF CASTING (BOTTOM AND TOP)

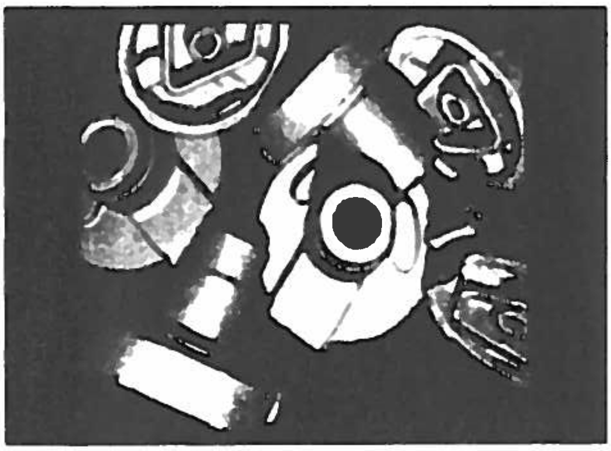

FIGURE 3 INTENSITY IMAGE OF A JUMBLE

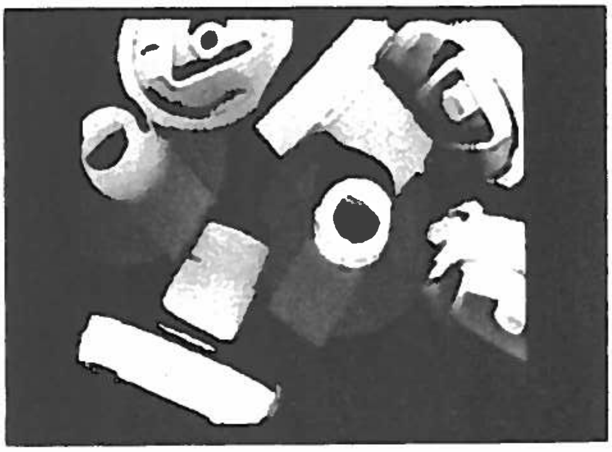

FIGURE 4 HEIGHT IMAGE OF A JUMBLE

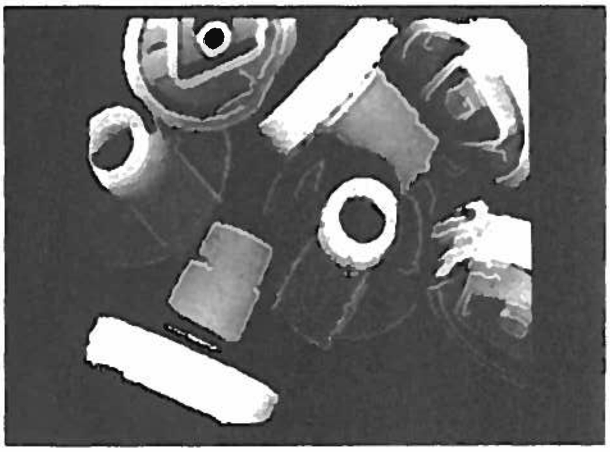

FIGURE 5 NON-SHADOW RANGE EDGES

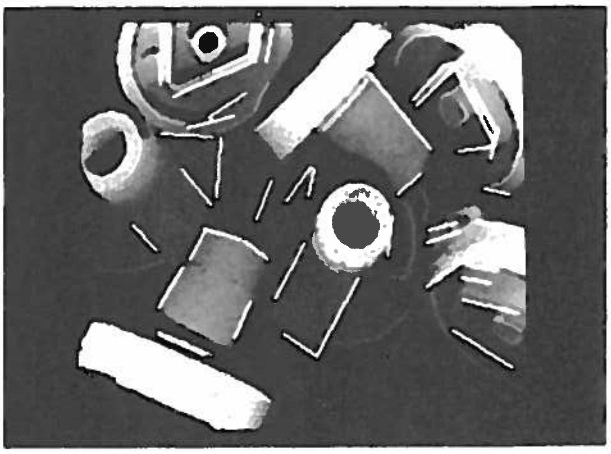

FIGURE 6 CIRCULAR ARCS AND STRAIGHT LINES

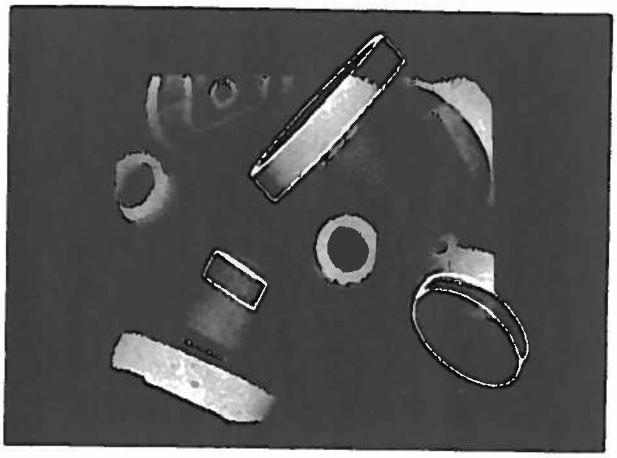

FIGURE 7 CYLINDERS DERIVED FROM TANGENTIAL EDGES

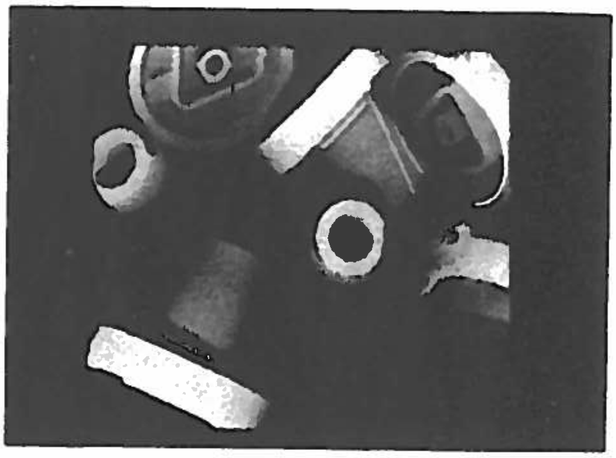

FIGURE 9 A CIRCLE FITTED TO AN ARC

NAME	RADIUS	PLANE	DIHEDRAL ANGLE	ARC ANGLE	LENGTH	PLANE WIDTH	CYLINDER WIDTH
Base-bottom	2.30	inside	90	360	14.4	0.4	1.0
Base-top	2.30	inside	90	317	12.7	1.3	1.0
Shelf-bottom	2.30	outside	-90	43	1.7	0.4	1.2
Inside-base	1.90	outside	90	360	11.9	0.4	0.7
Pipe-shoulder	1.00	inside	90	360	6.3	0.1	1.8
Pipe-top	0.90	inside	90	360	5.7	0.3	0.9
Pipe-base	1.00	outside	-90	243	4.2	1.3	1.8
Pipe-on-shelf	1.00	outside	-90	116	2.0	1.6	1.5
Pipe-joint	0.90	outside	-90	360	5.7	0.1	0.9
Corner-1	0.68	inside	90	100	1.2	0.2	0.7
Small-cylinder	0.50	inside	90	360	3.1	0.2	0.7
Corner-2	0.38	inside	90	89	0.6	0.2	0.7
Corner-3	0.33	inside	90	103	0.6	0.2	0.7
Inside-pipe	0.55	outside	90	360	3.5	0.3	2.6
Corner-4	0.43	outside	90	100	0.7	0.2	0.7

FIGURE 8 TABLE OF CIRCULAR ARCS

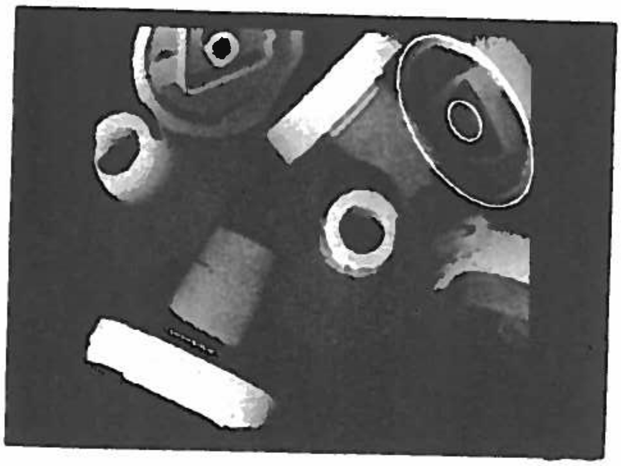

FIGURE 10 CONCENTRIC CIRCLES

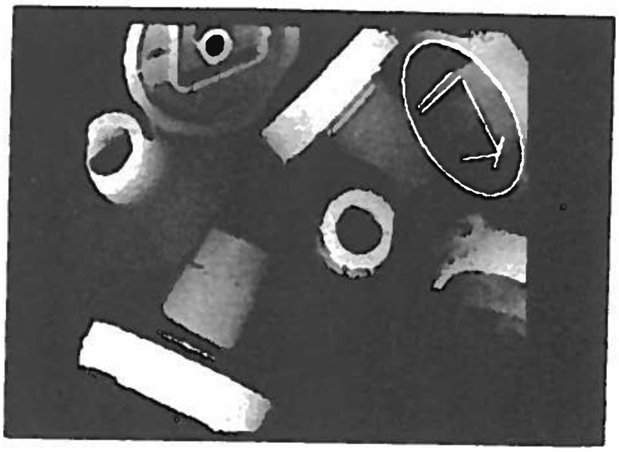

FIGURE 11 CANDIDATE LINES FOR DETERMINING THE ROTATION

FIGURE 12 HYPOTHESIZED CASTING

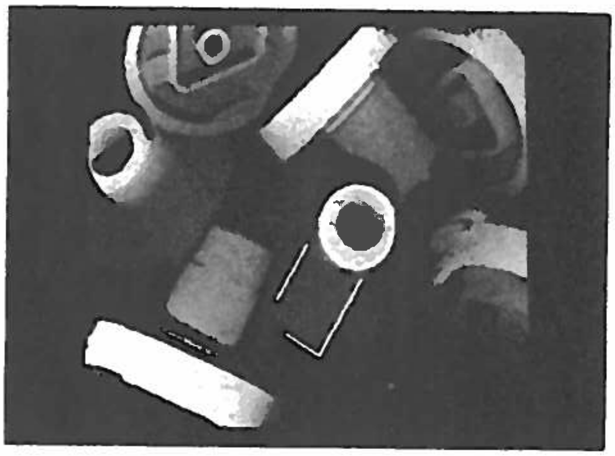

FIGURE 13 CIRCLE AND CANDIDATE LINES

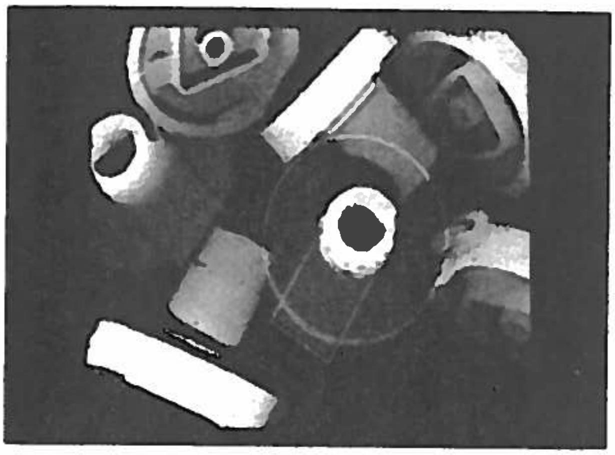

FIGURE 14 HYPOTHESIZED CASTING

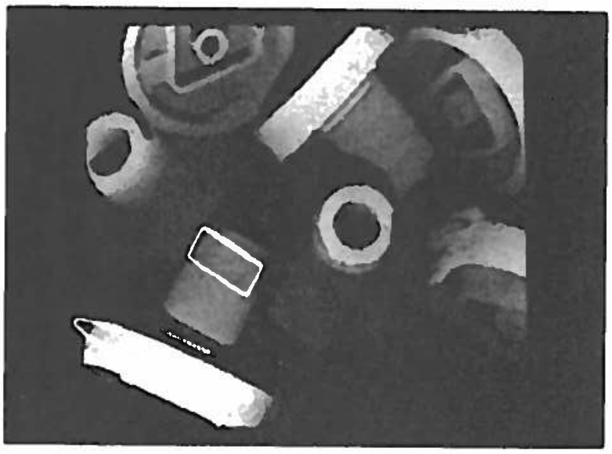

FIGURE 15 CYLINDER AND COMPATIBLE CIRCLE

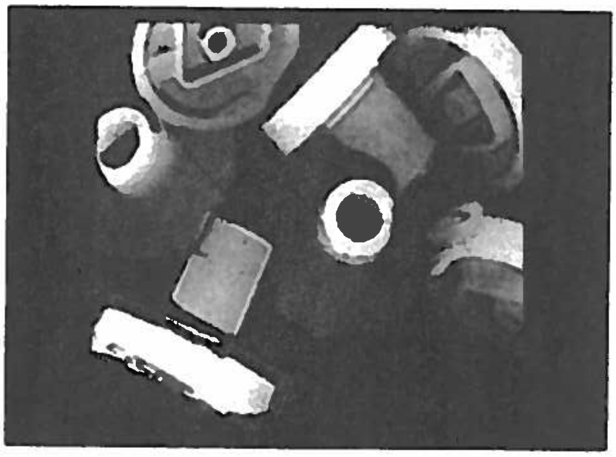

FIGURE 16 HYPOTHESIZED CASTING

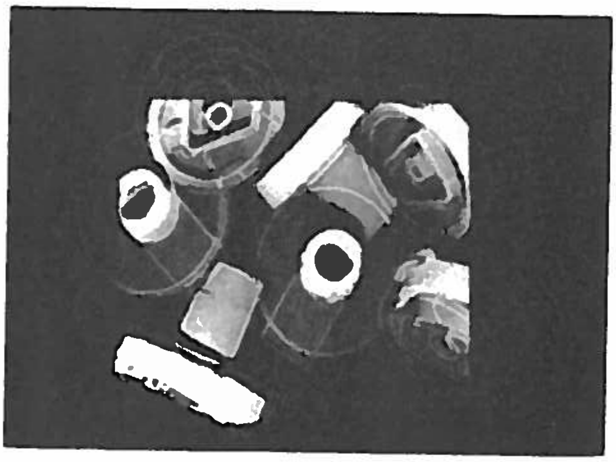

FIGURE 17 SIX HYPOTHESIZED CASTINGS