

3DPO: A Three-Dimensional Part Orientation System*

Robert C. Bolles, Patrice Horaud, and Marsha Jo Hannah

SRI, International
333 Ravenswood Ave.
Menlo Park, CA 94025

ABSTRACT

A system for recognizing and locating three-dimensional objects in range data is presented. Design considerations are discussed, followed by a description of a new object representation scheme, designed to facilitate object recognition. Experimental results are given for a preliminary implementation of the object modeling scheme and for an initial version of the object recognition system.

INTRODUCTION

One factor inhibiting the application of industrial automation is the lack of general methods for acquiring a part from unstructured storage so that it can be presented to the work station in a known position and orientation. A classic example of this limitation is the "bin-picking" problem--automating the job of a worker who reaches into a bin of jumbled parts and retrieves one part at a time, as needed. Solving this problem is difficult because a part in a bin may lie in infinitely many locations and may be partially occluded or buried.

Our research goal is to develop general-purpose techniques for recognizing and locating partially visible parts with three-dimensional (3-D) uncertainties in their locations. To achieve this goal, our approach is to develop 3-D part models that facilitate interpretation of range and intensity information on the basis of reasoning. Our rationale is that, first of all, range data simplify the locational analysis because the 3-D geometric information is encoded directly in such data. Second, familiarity with the model of a part will add enough new constraints to make it practical to recognize and locate different parts jumbled in a bin.

Most previous work in the area of range data processing has either concentrated on the recognition of simple, generic objects, such as polyhedra, cylinders, and spheres (e.g., see [1,2,3]) or on the description of a scene in terms of generic primitives, such as planar patches (e.g., see [4,5]) or in terms of generalized cylinders (e.g., see [6,7]). Only a few groups have investigated techniques for recognizing specific objects (e.g., see [8,9,10]). Previous work in bin picking has often pursued a two-step recognition and location strategy: (1) physically acquire a part from the bin, and (2) determine its part's location in the hand or on some intermediate holding place, such as a table (e.g., see [11,12]).

Our ultimate goal is to develop a program that can understand a configuration of parts well enough to determine the location of a part and direct an arm to grasp it, then predict which parts will move when the selected one is lifted out of the pile. This approach will result in a one-step process in which the arm system knows the approximate location of each part when it picks it up. To reach

this goal, we are simultaneously exploring ways to represent specific features of parts in order to facilitate the recognition process, while developing a recognition procedure that capitalizes on the information available in these extended parts models.



FIGURE 1 CASTING TO BE LOCATED

We are considering moderately complex parts, such as the casting in Figure 1, for two reasons. First, they are typical of a large class of industrial parts. Second, we expect (and have found) the abundance of features in a complex part to be helpful rather than confusing because distinctive clusters of local features can be used to recognize the part from a partial view of it. For example, two concentric circles connected by a planar surface strongly suggest the end of a thick-walled pipe, such as the portion of the casting in Figure 1.

One of our basic assumptions is that hypothesis generation and matching should be based on a few (e.g., 2 or 3) features or feature clusters rather than on many (e.g., 30) features, each with several (e.g., 5 to 10) possible interpretations. Using this approach, almost any matching algorithm, even an exhaustive search, can efficiently determine the best match. The problem is to be able to find those few features that are sufficient to distinguish the part. The solution to this problem is based on feature representation and preliminary planning.

The use of a small number of feature clusters is also appropriate because each feature provides a substantial

* The work reported herein was supported by the National Science Foundation under grant DAR-7922248.

amount of information. For example, the position and orientation of a dihedral edge determines all but one (position along the edge) of the six degrees of freedom associated with an object; the position and orientation of the end of a cylinder also determines all but one degree of freedom (rotation around its axis).

The function of the preliminary planning is to enumerate the occurrences of feature clusters and evaluate their utility so that the execution system can efficiently use each detected feature. This goal-oriented planning system may even provide additional information, such as a list of nearby features that may help identify a feature. A significant amount of computer time performing this type of analysis is justified because it is only done once and because it increases the efficiency of the run-time system, which is executed many times.

One goal of our research is to develop a program that automatically converts computer-aided-design (CAD) models of 3-D parts into a recognition-based representation that will help the system recognize and locate the parts. To achieve this goal we expect to extend the techniques we developed for the analysis of two-dimensional (2-D) parts [13].

We have so far concentrated on the representation of 3D objects and on the detection of feature clusters in range data. In this paper we first outline the recognition process and then briefly describe our progress in the development of a representation and the detection of feature clusters.

THE 3D PART ORIENTATION (3DPO) SYSTEM

We partitioned the recognition process into five steps:

- (1) Primitive feature detection
- (2) Feature cluster formation
- (3) Hypothesis generation
- (4) Hypothesis verification
- (5) Parameter refinement.

For the first step we envision a combination of special-purpose hardware that performs the computationally intensive low-level detection of primitive features, such as discontinuities, and software that locates object-specific features, such as a circular arc with a specific radius. The hardware analysis will be applied to all images in a uniform way except for possibly changing a few parameters for different types of parts. The software processing involves the application of general-purpose techniques that use lists of part features to obtain specific feature parameters.

The purpose of the cluster formation step is to "grow" clusters of features that can be used to hypothesize the type, position, and orientation of a part. In this step a feature cluster is grown by locating features that are directly related to ones in the cluster. Each feature in a part model has a list of related features. The types of relationships include "topologically connected" and "in the same plane."

In the third step, one or more clusters of features are used to generate hypotheses about the part types and locations. If individual clusters are sufficient, they are used; otherwise, a strategy similar to the one used in the local-feature-focus method [13] is used to recognize clusters of clusters. Note that this process involves more searching than the cluster formation step because there are no relationships between the clusters except their relative positions that can be used to suggest good candidates for extension. It is this step in the recognition process that must be minimized to avoid the problems of combinatorial explosion.

The purpose of the fourth step is to check a hypothesis by looking for additional features that are consistent with it. There are two ways to do this: one is to use the hypothesis to predict the locations of features to be found; the second is to see if the features already detected agree with the hypothesis. The first way is sounder, but it requires a prediction system that includes computationally expensive operations, such as hidden-line elimination. Finally, given a verified hypothesis, the fifth step applies an averaging technique, such as a least-squares, to compute a more precise location of the part.

In the following sections we briefly describe the part-modeling system and techniques for locating feature clusters. Throughout this discussion, we use examples from the recognition of the casting shown in Figure 1 in range data obtained by a projector-camera range sensor [14]. Figure 2 shows a perspective image of the 3-D intersections of the sensor's light plane with two castings of the type shown in Figure 1, one up-side-down relative to the other.

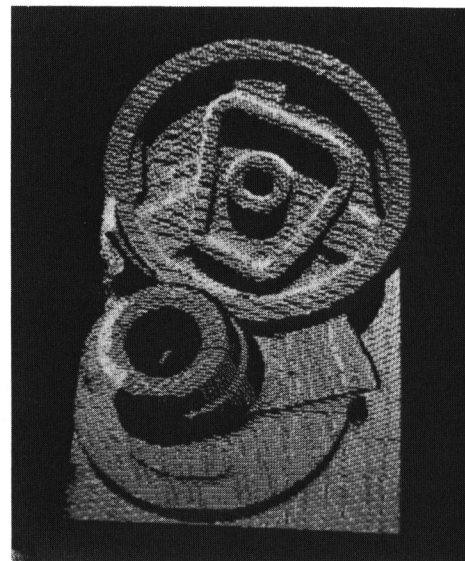


FIGURE 2 PERSPECTIVE IMAGE OF PLANAR LIGHT INTERSECTIONS WITH TWO CASTINGS (137 SLICES WITH 240 POINTS PER SLICE)

OBJECT MODELS

CAD systems and their representations are designed for constructing and displaying parts, not for recognizing them. Even though a CAD model may be complete in the sense that it contains all the 3-D information about a part, there are more convenient representations for a recognition system than a CAD model. For example, a CAD model might state the size and position of a hole, but a recognition procedure needs a list of all holes that size. In general, a recognition system must be able to provide answers quickly to such questions as:

- How many features are there of a given type and size?
- Which surfaces intersect to form this edge?
- What other features lie in this plane?
- What neighboring features could be used to distinguish this feature from others like it?

To answer these questions efficiently, each part feature should be listed under several different classifications. For example, a dihedral edge should be in the lists of edges bounding the two surfaces that meet and form the edge; it should be in the list of all dihedral edges (classified according to their included angles); and it may also be in other lists, such as the list of features in a specific plane. This redundancy is the key to efficient recognition.

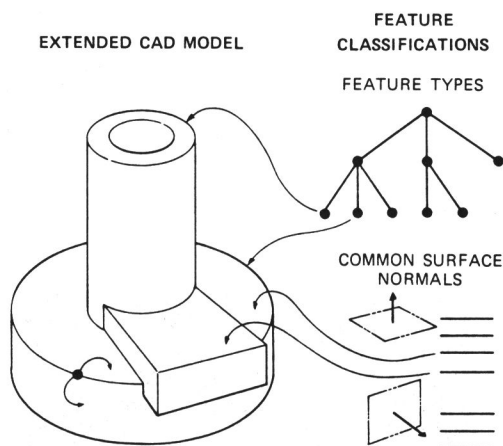


FIGURE 3 MODELING A 3-D PART BY EXTENDING ITS CAD MODEL AND CLASSIFYING ITS FEATURES

In the 3DPO system the model of a part consists of two components—an extended CAD model and a set of feature classification networks (see Figure 3). The extended CAD model contains a standard volume-surface-edge-vertex description as well as pointers linking topologically connected features. For example, each edge points to the two surfaces that form it, and each surface contains an ordered list of its boundary edges and a list of its holes. We have implemented a preliminary version of this type of model, using a pointer structure similar to Baumgart's winged-edge representation [15].

The feature networks in the current implementation classify the features according to their types and sizes. We plan to include other groupings, such as lists of surface elements that share a common normal or lists of cylinders that share a common axis. Each representation will be designed for a set of special-purpose procedures that analyze the data in terms of one property, and yet all the representations will contain pointers back to the extended CAD model, which will serve as the core representation.

We implemented a set of routines that can answer some of the questions mentioned earlier by analyzing the topology of a part and extracting entries from the classification lists. For example, the current system can quickly produce a list of all circular edges that are concave dihedral angles and have radii between .8 and 1.0 inch.

In this modeling system we differentiate between view-independent relationships, such as topological connectivity, and view-dependent relationships, such as image proximity (in pixels). The view-independent relationships are functions of the inherent geometric characteristics of the part, such as its size and topology, and can easily be enumerated by an analysis of the part model. The view-dependent relationships, on the other hand, are functions

of the sensor, its type, and its location; they are significantly more difficult to enumerate. Even a simple part can have fifty or more structurally different views [16]. We plan to investigate techniques similar to those used in ACRONYM [17] for representing classes of similar views.

FEATURE DETECTION AND HYPOTHESIS GENERATION

The current, partial implementation of the 3DPO system can detect features, form simple clusters, and hypothesize objects. The system has two methods for detecting edges in range data. One method analyzes the discontinuities occurring in slices through the scene, and the other applies a zero-crossing edge detection method to a "height" image (e.g., see [18]). In the first method each discontinuity is labelled according to its geometric and photometric significance and then the labelled discontinuities are linked together. Figure 4 illustrates the types of discontinuities used in this method. The position of the light source is used to distinguish an object edge that casts a shadow from the edge of that shadow. Figure 5 shows the classified discontinuities in one slice of the data shown in Figure 2. Figure 6 shows the chains of discontinuities found in Figure 2 by the line-by-line linking process. Figure 7 shows only the chains that may correspond to physical edges in the scene. These are the chains that the program later tries to identify as specific object features.

The second method of detecting edges treats the range data in Figure 2 as a gray-scale image, locates zero crossings, and then filters out the insignificant ones (i.e., those that are too weak or too short). Figure 8 shows a grey scale version of the data in Figure 2, and Figure 9 shows the zero crossings detected at one resolution that passed the significance tests.

After the edges have been found, the system tries to identify object-specific features, such as a cylinder with a radius of .9 inches, by performing the following operations: (1) partitioning each edge into sequences of points that lie in a single plane; (2) partitioning each planar sequence into linear segments and circular arcs whose radii match radii in the part models; (3) characterizing the surfaces that intersect to form each segment in terms of their types, approximate sizes, and the relative angle between them; and (4) constructing a list of model features that match the characteristics of each observed segment. Figure 10 shows a plane fitted to a sequence of points and the position of the matching cylindrical body from the model. Figure 11 shows all the circular features found in Figure 2. Each one matches only one or two model features.

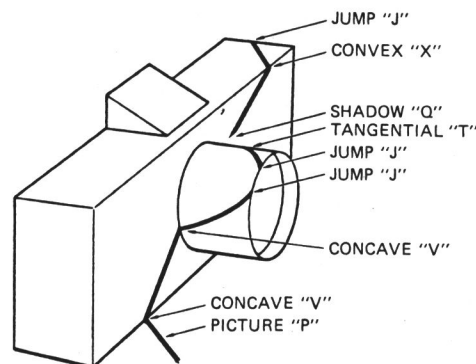


FIGURE 4 DISCONTINUITY CLASSIFICATIONS ALONG A SLICE

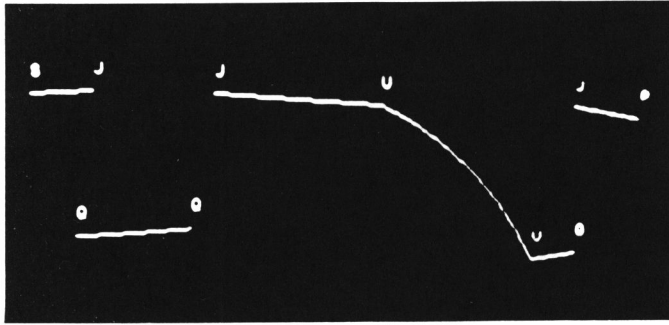


FIGURE 5 CLASSIFIED DISCONTINUITIES IN ONE SLICE

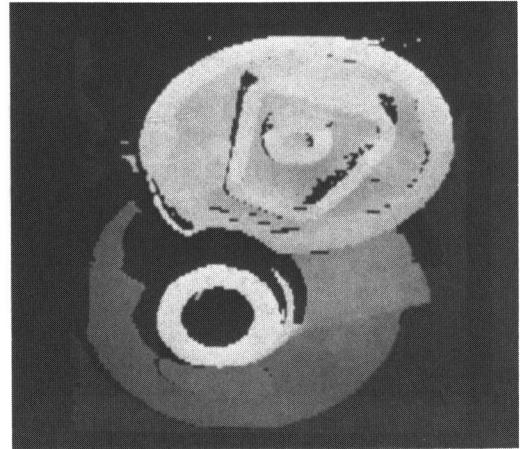


FIGURE 8 AN INTENSITY IMAGE IN WHICH THE BRIGHTNESS CORRESPONDS TO THE HEIGHT ABOVE THE TABLE

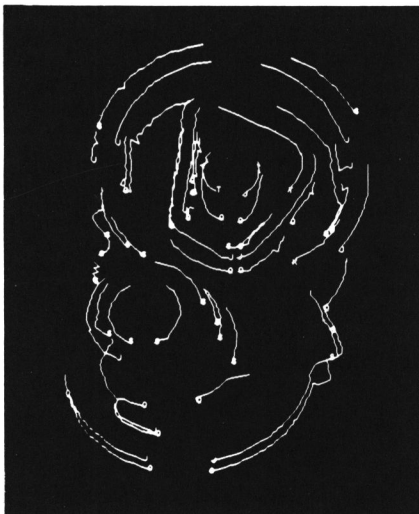


FIGURE 6 CHAINS OF DISCONTINUITIES FOUND IN FIGURE 2

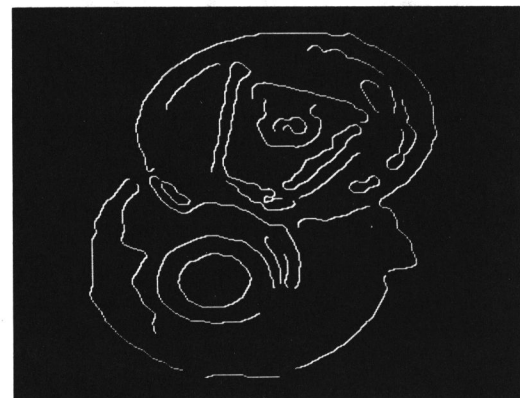


FIGURE 9 FILTERED ZERO CROSSINGS



FIGURE 7 CHAINS THAT MAY CORRESPOND TO PHYSICAL EDGES

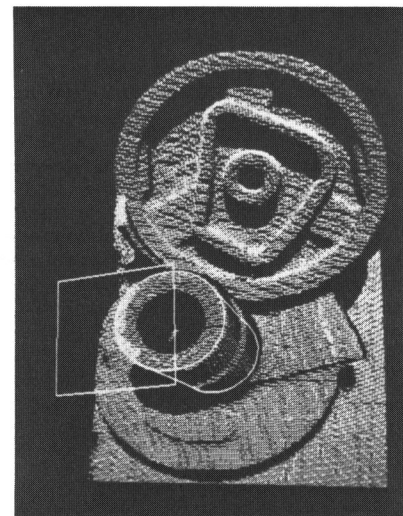


FIGURE 10 CYLINDER IMPLIED BY ONE EDGE

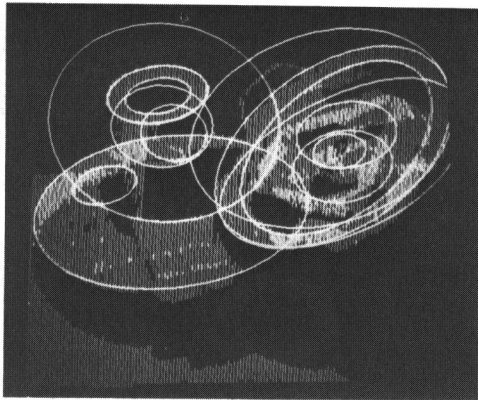


FIGURE 11 POSSIBLE ENDS OF CYLINDRICAL FEATURES

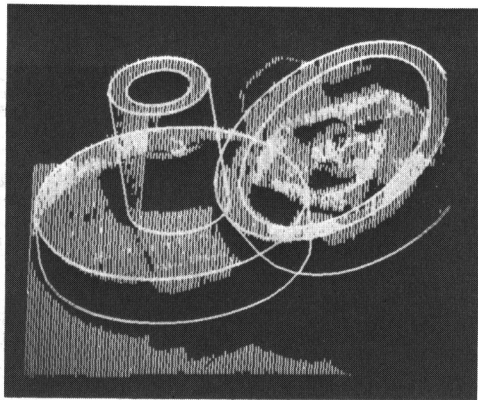


FIGURE 12 TWO HYPOTHESIZED CASTINGS

To hypothesize objects, the program searches for sets of mutually consistent features by examining all pairs of features. Figure 12 shows two hypothesized objects, one based on three cylinders and one based on two cylinders.

DISCUSSION

We have described a system for locating complex parts in range data and have shown the results of some of its components. We have argued for a recognition strategy that concentrates on a few feature clusters and a planning process that selects the best features to use. We have only partially demonstrated the effectiveness of this approach. We plan to improve the 3DPO system and continue the comparison of matching strategies. We also plan to explore the use of an existing CAD system, such as PADL-2 [19], as the basis for a more complete implementation of our representation scheme.

REFERENCES

[1] Shirai, Y., and M. Suwa, "Recognition of Polyhedrons with a Range Finder," *Proc. of the Second IJCAI*, pp. 80-87, London, England (August 1971).

- [2] Popplestone, R.J., C.M. Brown, A.P. Ambler, and G.F. Crawford, "Forming Models of Plane-and-cylinder Faceted Bodies from Light Stripes," *Proc. of the Fourth IJCAI*, Tbilisi, Georgia, USSR, pp. 664-668 (September 1975).
- [3] Nitzan, D., A.E. Brain, and R.O. Duda, "The Measurement and Use of Registered Reflectance and Range Data in Scene Analysis," *Proc. of the IEEE 65*, pp. 206-220 (1977).
- [4] Oshima, M., and Y. Shirai, "A Scene Description Method Using Three-Dimensional Information," *Pattern Recognition*, Vol. 11, pp. 9-17 (1978).
- [5] Faugeras, O.D., et al, "Toward a Flexible Vision System," *Robot Vision*, Alan Pugh, editor (IFS Publications, Ltd., United Kingdom, 1983).
- [6] Agin, G.J. and T.O. Binford, "Computer Description of Curved Objects," *Proc. of the Third IJCAI*, Stanford University, Stanford, California, pp. 629-640 (August 1973).
- [7] Nevatia, R., and T.O. Binford, "Description and Recognition of Curved Objects," *Artificial Intelligence*, Vol. 8, pp. 77-98 (1977).
- [8] Shneier, M., "A Compact Relational Structure Representation," *Proc. of the Sixth IJCAI*, pp. 818-826, Tokyo, Japan (August 1979).
- [9] Sugihara, K., "Range-data Analysis Guided by a Junction Dictionary," *Artificial Intelligence 12*, pp. 41-69 (1979).
- [10] Oshima, M., and Y. Shirai, "Object Recognition using Three-Dimensional Information," *Proc. of the Seventh IJCAI*, University of British Columbia, Vancouver, B.C., Canada, pp. 601-606 (August 1981).
- [11] Rosen, C.A., and D. Nitzan, et al., "Exploratory Research in Advanced Automation," Fourth Report, NSF Grant G138100X1, SRI Project 2591, Stanford Research Institute, Menlo Park, California (June 1975).
- [12] Birk, J.R., R.B. Kelley, and J. Dessimoz, "Visual Control for Robot Handling of Unoriented Parts," *Proc. of the SPIE Technical Symposium East*, Vol. 281, Washington, D.C. (April 1981).
- [13] 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).
- [14] Bolles, R.C., J.H. Kremers, and R.A. Cain, "A Simple Sensor to Gather Three-Dimensional Data," A.I. Center Technical Note 249, SRI International, Menlo Park, California (July 17, 1981).
- [15] Baumgart, B.G., "Winged Edge Polyhedron Representation," Stanford A.I. Memo AIM-179, Stanford University, Stanford, California (1972).
- [16] Chakravarty, I., "The Use of Characteristic Views as a Basis for Recognition of Three-dimensional Objects," Ph.D. thesis, Rensselaer Polytechnic Institute (October 1982).
- [17] Brooks, R.A., "Symbolic Reasoning Around 3-D Models and 2-D Images," *Artificial Intelligence Journal*, Vol. 17, pp. 285-348 (1981).
- [18] Marr, D., and E.C. Hildreth, "Theory of Edge Detection," *Proc. R. Soc. Land. B*, 207, pp. 187-217 (1980).
- [19] Brown, C.M., "Some Mathematical and Representational Aspects of Solid Modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-3, No. 4, pp. 444-453 (July 1981).