CAD-BASED 3-D OBJECT RECOGNITION[1]

Chuck Hansen, Thomas C. Henderson and Rod Grupen*

Department of Computer Science, The University of Utah
COINS*, University of Massachusetts

Abstract

We propose an approach to 3-D object recognition using CAD-based geometry models for freeform surfaces. Geometry is modeled with rational B-splines by defining surface patches and then combining these into a volumetric model of the object. Characteristic features are then extracted from this model and subjected to a battery of tests to select an "optimal" subset of surface features which are robust with respect to the sensor being used (e.g. laser range finder versus passive stereo) and permit recognition of the object from any viewing position. These features are then organized into a "strategy tree" which defines the order in which the features are sought, and any corroboration required to justify issuing a hypotheses. We propose the use of geometric sensor data integration techniques as a means for formally selecting surface features on free-form objects in order to build recognition strategies. Previous work has dealt with polyhedra and generalized cylinders, whereas here we propose to apply the method to more general surfaces.

1 Introduction

Our goal is to develop a systematic approach for both the generation of representations and recognition strategies based on CAD models. Such a system provides an integrated automation environment and permits automatic process planning. The system is composed of several components: a CAD system, a milling system, a recognition system and a manipulation system. In previous work, we have explored the automatic generation of recognition strategies based on the CAGD model[7,8,10,11]. There we use the shape representation inherent in CAGD models to drive the recognition process. Others have been studying portions of this system. Recent work by Ho has focused on the generation of computer vision models directly from a CAGD model[1,12].

Strategy trees provide a robust mechanism for recognition and localization of three dimensional objects (occluded as well as non-occluded) in typical manufacturing scenes. Whereas our earlier work dealt with the use of geometric knowledge in constructing *strategy trees* for polyhedra and swept surfaces, here we examine feature selection for free-form surfaces.

1.1 Related Work

One of the first researchers to study the automatic synthesis of general recognition strategies was Goad[5]. He was concerned with automatic programming for 3-D model based vision. His work generated a recognition scheme for matching edges based on a general sequential matching algorithm. His algorithm proceeded in three steps: (1) predict a feature, (2) observe (match) a feature, and (3) back-project (refine the object hypothesis based on step 2). These three steps form a template which is used by the automatic programming phase. He used a unit sphere to gather *loci* of viewangles (camera positions) which represent orientations of the object. His work differs from that described here in that he obtained 3-D interpretations of 2-D intensity images rather than 3-D sensor data. The only features used were straight edges from intensity images and the search trees were generated from a template and ordered by hand rather than automatically. His system did not consider partial occlusion. However, this was a major contribution since it was one of the first attempts to automate the generation of recognition schemes.

Another influential project was the 3DPO system by Bolles and Horaud[3]. This work is the 3-D generalization of the Local Feature Focus method[2]. Their system annotates a CAD model producing what is called the extended CAD model. From this model, feature analysis is performed to determine unique features from which to base hypotheses. The focus feature in their system is the dihedral arc. When the recognition system finds a dihedral arc, it looks for nearby features which are used to discriminate between model arcs with similar attributes. From these, an object's pose is hypothesized and subsequently verified. The work here work closely parallels the 3DPO system. However, focus features were hand chosen in 3DPO as were the local features used for discrimination.

Recently, Ikeuchi has explored the use of *interpretation trees* for representation of recognition strategies[13]. His system uses the concept of visible faces to generate generic representative views, called aspects. From this set of aspects, an interpretation tree is formed which discriminates among

[1]This work was supported in part by NSF Grant IRI-8802585 and DARPA Contract N00014-88-K-0688. All opinions, findings, conclusions or recommendations expressed in this document are those of the author and do not necessarily reflect the views of the sponsoring agencies.

CH2809-2/89/0000-0168 $1.00 © 1989 IEEE

the different aspects. His system uses a variety of object features such as: EGI, face inertia, adjacency information, face shape, and surface characteristics. Most of these features are based on planar faces. A very specific interpretation tree is generated for an object using a set of object specific rules. The rules were selected by hand rather than generated automatically. There does not appear to be any algorithmic approach for the application of the rules to discriminate between the aspects. The branching on the tree seems to be a function of the particular aspects chosen rather than being based on the geometric information in the model.

1.2 Strategy Trees

The system developed in this paper is not dependent on a certain class of features but rather can be extended to include many classes of features not implemented at this time. The system also performs automatic selection of features based on a set of constraints: feature filters. These features are used to form a *strategy tree* which provides a scheme for hypothesis formation, corroborating evidence gathering and object verification. Given a CAD model of an object, a specific, tailor-made system to recognize and locate the object is synthesized.

To attain this goal, the following problems have been solved:

1. **Geometric Knowledge Representation**: The use of geometric data is central to a strong recognition paradigm. Weak methods can only be avoided when better information is available. The Alpha_1 B-spline model allows the modeling of freeform sculptured surfaces. To obtain the geometric features of interest for 3-D recognition, techniques for the transformation to a computer vision representation have been developed.

2. **Automatic Feature Selection**: The part to be recognized or manipulated must be examined for significant features which can be reliably detected and which constrain the object's pose as much as possible. Moreover, such a set of features must *cover* the object from any possible viewing angle. In solving the feature selection problem, a technique is available for synthesizing recognition systems. This produces much more efficient, robust, reliable and comprehensible systems.

3. **Strategy Tree Synthesis**: Once a robust, complete and consistent set of features has been selected, a search strategy is automatically generated. Such a strategy takes into account the strongest features and how their presence in a scene constrains the remaining search. The features and the corresponding detection algorithms are welded, as optimally as possible, into a search process for object identification and pose determination. The automatic synthesis of search strategies is a great step forward toward the goal of automated manufacturing. Generation of strategies is constrained, not only by the feature selection process but, by the actual task to be accomplished. Thus, strategies for a specific task might not be as strong when applied to a different task; strategies are task specific.

In this paper we describe how techniques from geometric data sensor integration can be used to select features in freeform surfaces.

2 Automatic Feature Selection

Several kinds of knowledge are required for feature selection. Geometric knowledge permits the selection of a complete and consistent set of features, while the sensor knowledge provides information on the robustness and reliability with which such features can be extracted. On the other hand, domain specific information about the task can be used to select feature extraction algorithms based on their complexity, robustness, etc.

The feature selection process can be viewed as a set of *filters* applied to the complete original set of features of an object. Filters select and rank features; order of application is important. Conceptually, the filters remove features from the input, in order of application, which do not meet the filter's criteria. The goal here is to automate and optimize this filtering process. The filters select features based on the following qualities:

- **rare** - histogram the features; rare features are useful for quickly identifying the object; these features make good root nodes in a search tree.
- **robust** - measure of how well the features can be detected; error and reliability.
- **cost** - measure of complexity (space and time) for computing feature.
- **complete** - does set of features cover all possible views of the object.
- **consistency** - how completely does feature characterize object pose; (i.e., how many DOFs are unresolved); how well does the feature differentiate between objects; measure of likelihood of correctly identifying the object.

When used in combination, these filters provide the mechanism with which to build a strategy tree. The task requirements may be such that the result of these filters is the null set of features. This can be dependent on the order in which the filters are applied to the complete feature set. For example, if the filter for rare features determines that a 1/4 inch dihedral edge is the *best* feature and is applied prior to the robustness filter, that dihedral might not be accepted by the robustness filter since it is so small. Thus, the set of features would be null after the application of the robustness filter. Whereas, if the robustness filter is applied first, it wouldn't accept such features and when the rare filter is applied to the features accepted by the robustness filter, it would determine a different set of features as being *best*. The order of application is to be determined by knowledge of both the task to be accomplished and experience; this aspect of our approach merits further study.

Since there is this possibility of null feature sets when filters are applied such that they absolutely eliminate features, the filters need to be applied in a relative manner. That is, the filters should rank the features rather than just eliminate those which don't meet the criteria. If the features are ranked by the filters, null sets should never occur. However, the order of application is still important.

A strategy tree is a generalization of a hierarchical classifier or decision tree and describes the search strategy used to recognize and determine the pose of objects in a scene. The use of strategy trees permits one to exploit knowledge of relations between the geometric features in the models. Such trees also define a sequence of measurements or evaluations of the scene data so as to eliminate certain classifications at particular nodes.

The system consists of two parts: the off-line model analysis and strategy generation and the run time environment. The CAD model is analyzed in terms of the geometric knowledge needed for object recognition. This geometric information, which is analyzed by the feature selection process, is used by the strategy tree builder to produce the core of the run time recognition system. During run time, the strategy tree provides the search structure and control for the hypothesis generator. By using the information provided from the feature extractors and the strategy trees, the hypothesis generator attempts to hypothesize pose descriptions for recognized objects in the scene. These hypotheses are verified for correctness and a description of recognized objects and their poses are the end result.

The matching strategy consists of two phases: the hypothesis generation phase and the hypothesis verification phase. This recognition technique is known as hypothesize and verify. The hypothesis generation phase is controlled by the strategy tree and the verification phase substantiates or refutes the hypotheses generated from the strategy tree.

A strategy tree consists of three major parts:

1. <u>The Root</u> - Which represents the object to be recognized.
2. <u>Level 1 Features</u> - Which are the strongest set of view independent features chosen for their ability to permit rapid identification of the object and its pose.
3. <u>Corroborating Evidence Subtrees, CES</u> - Whose purpose is twofold: they direct the search for corroborating evidence that supports the hypothesis of the level 1 features and they direct the search for geometric information to completely determine the pose prior to hypothesis generation.

Strategy trees determine the procedure a recognition system follows for object recognition. There will be at least one strategy tree for each model under consideration. If a model is used in a different task or environment, there could possibly be a different strategy tree for each of those tasks. The level 1 features are selected using the *feature filters*. These conform to the requirements which constrain the task, environment, and model yet contain the strongest geometric information which leads to a solution. The corroborating evidence subtrees, CES, are constructed using geometric information derived from the CAD model.

The strategy tree *guides* the search through possible solutions. When a level 1 node is matched in the strategy tree and it is supported by the Corroborating Evidence Subtrees, then a hypothesis is generated. The hypothesis is passed to an object verifier which determines whether the hypothesis is valid within some confidence level.

Once a level 1 node has been matched using the heuristics described above, and a determination made as to whether the feature is occluded or not, the local CES can be evaluated, as prescribed by the strategy tree. This local evidence gathering limits the number of hypotheses generated and passed to the object verification phase by determining whether a hypothesis is justified by the local evidence. If there isn't supporting local evidence, as prescribed by the strategy tree, then that level 1 match fails and the detected feature is marked as unmatched. If there is enough local supporting evidence, a hypothesis is generated for the object verification phase to accept or reject.

Two forms of verification have been examined: structural and pixel correlation. Structural verification refers to verifying spatial relations among the features which should be present in the scene. This is similar to relational graph matching in 2-D. For an example of this type of strategy, see[9].

Pixel correlation refers to the verification technique of matching predicted depth, pixel by pixel, in a generated image and the sensed *image*. This corresponds to template matching in 2-D and requires a good camera model.

Either of these methods provides for verification. This follows the hypothesis verification techniques used by others[2 3,14]. One of three states is assigned to the match of the hypothesized feature or pixel with the observed feature or pixel:

- **positive evidence** When the observed feature or depth is approximately the same as predicted. This means the observed object matches the transformed model in the predicted image.
- **neutral evidence** When the observed feature or depth is closer to the sensor than the predicted one. This seems counterintuitive but it simple means that the predicted feature/depth can't be observed because some thing is possibly blocking sight of the object. In the presense of occlusion, it can't be determined whether the difference between the prediction and the scene is due to an incorrect hypothesis or due to an occluding object. This also holds for shadow pixel/region in the range image for the same reason.
- **negative evidence** When the observed feature or depth is much farther from the sensor than the predicted one. This definitely points to an incorrect hypothesis since the observed feature/depth is not occluded but is not where it should be.

If these measures are accumulated for the predicted range image or structural features, the hypothesis can be quantified and accepted or rejected accordingly. This quantification provides a measure of confidence in the hypothesis.

3 Geometric Sensors for CAD Feature Selection

The concepts which have been outlined above have been implemented in an experimental system. The equipment used for the experiments consists of a Technical Arts 100A White Scanner, DEC VAX class processors and an HP Bobcat. Feature computation was coded on a VAX 750 in C. The automatic generation of strategy trees and the matcher were coded on an HP Bobcat in HP Common Lisp.

Range data was obtained with the White Scanner 100A which returns actual Cartesian data. The structured light is a laser beam which is spread into a plane of light and directed onto the work space. The sensing mechanism is a GE CCD camera with a 240 x 240 image.

The notion of a "geometry sensor" has been introduced in the robotics literature[4] as well as several other approaches to this idea[6,15]. A sensor is viewed as a device which provides parameters which characterize some class of geometric element (e.g., a point, a line, a plane). This model permits the characterization of error in the sensed data, as well as the determination of the error in both data which undergoes a transformation of coordinate frames, and in higher-level features produced from sensed data (such as lines from points).

In selecting surface features for a strategy tree to recognize a given object, we need some strong criteria for selecting patches of the surface of the object (in terms of their curvature). We propose geometric data integration methods as a formal approach to the selection of surface features in free-form surfaces. That is, given a model of the sensor (for example, a laser range finder), it is possible to determine the reliability and robustness of the curvature features of a particular region of the surface of the object under consideration by determining the reliability of higher-level surface curvature features derived from the sensed data. We are currently exploring the implementation of this idea.

4 Conclusions and Future Work

It has been shown that the automatic generation of recognition strategies is possible. A method is presented which analyzed the geometric information of an object to determine the best strategy for recognition within the constraints of the sensing environment and the task. Using this information, a recognition system, a strategy tree, is produced which effectively matches models with sensed data. The strategy tree generation is performed automatically with minimal assistance from the user. The strategy tree provides a model based approach for the recognition and location of objects using 3-D sensing techniques. These strategy trees are formed using the following feature filters: robust, complete, consistent, unique, and cost effective. Using these filters, a strategy is formed which includes the use of corroborating evidence to substantiate hypotheses at formation time thereby increasing the speed for recognition.

We have proposed the use of geometric data integration techniques as a means to have formal criteria for the selection of surface curvature features in free-form objects. That is, a good sensor model is required as well as the analysis of the robustness of features derived from sensed geometric data.

References

[1] B. Bhanu and C.C. Ho. CAGD-Based 3-D Object Representations for Computer Vision. *IEEE Computer*, 20(8):19–36, August 1987.

[2] R.C. Bolles and R.A. Cain. Recognizing and Locating Partially Visible Objects: The Local-Feature-Focus Method. *Robotics Research*, 1(3):57–82, 1982.

[3] R.C. Bolles and P. Horaud. 3DPO: A Three-Dimensional Part Orientation System. *Robotics Research*, 5(3):3–26, 1986.

[4] H. F. Durrant-Whyte. Consistent Integration and Propagation of Disparate Sensor Observations. *International Journal of Robotics Research*, 6(3):3–24, 1987.

[5] C. Goad. Special Purpose, Automatic Programming for 3D Model-Based Vision. In *Proceedings of the DARPA Image Understanding Workshop*, pages 94–104, DARPA, 1983.

[6] Greg Hager and Max Mintz. Searching for Information. In Avi Kak and Su-shing Chen, editors, *Proceedings of the Workshop on Spatial Reasoning and Multi-Sensor Fusion*, pages 313–322, October 1987.

[7] C.D. Hansen. *CAGD-Based Computer Vision: The Automatic Generation of Recognition Strategies.* PhD thesis, The University of Utah, Salt Lake City, Utah, July 1988.

[8] Charles D. Hansen and Thomas C. Henderson. CAGD-Based Computer Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11:to appear, 1989.

[9] T.C. Henderson, Chuck Hansen, Ashok Samal, C.C. Ho, and Bir Bhanu. CAGD Based 3-D Visual Recognition. In *Proceedings of the International Conference on Pattern Recognition*, pages 230–232, Paris, France, October 1986.

[10] Thomas C. Henderson, C. Hansen, R. Grupen, and B. Bhanu. *The Synthesis of Visual Recognition Strategies.* Technical Report UUCS-TR-102, The University of Utah, Department of Computer Science, September 1986.

[11] Thomas C. Henderson, E. Weitz, C. Hansen, R. Grupen, C.C. Ho, and B. Bhanu. CAD-Based Robotics. In Art Sanderson, editor, *Proceedings of the IEEE Conference on Robotics and Automation*, pages 631–635, Raleigh, NC, May 1987.

[12] C.C. Ho. *CAGD-Based 3-D Object Representations for Computer Vision.* Master's thesis, University of Utah, Salt Lake City, Utah, June 1987.

[13] K. Ikeuchi. Model-Based Interpretation of Range Imagery. In *Proceedings of the DARPA Image Understanding Workshop*, pages 321–339, DARPA, 1987.

[14] T. Knoll and R. Jain. Recognizing Partially Visible Objects using Feature Indexed Hypotheses. *J. of Robotics and Automation*, RA-2(1):3–13, 1986.

[15] R. Smith, M. Self, and P. Cheeseman. Estimating Uncertain Spatial Relationships in Robotics. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 850–850, Raleigh, NC, 1987.