Scene Understanding

Analysis of Scenes Containing Multiple Non-polyhedral 3D Objects

Mauro S. Costa[1] and Linda G. Shapiro[1,2] *

[1] Department of Electrical Engineering, FT-10
[2] Department of Computer Science & Engineering, FR-35
University of Washington
Seattle WA 98195
U.S.A.

Abstract. Recognition of generic three-dimensional objects remains an unsolved problem. Scenes containing multiple nonpolyhedral 3D objects are particularly challenging. Conventional object models based on straight line segments and junctions are not suitable for this task. We have developed an appearanced-based 3D object model in which an object is represented by the features that can be most reliably detected in a training set of real images. For industrial objects with both flat and curved surfaces, holes, and threads, a set of useful features has been derived; and a recognition system utilizing these features and their interrelationships is being developed. The recognition system uses small relational subgraphs of features to index the database of models and to retrieve the appropriate 3D models in a hypothesize-and-test matching algorithm. This paper describes the new models, the matching algorithm, and our preliminary results.

1 Introduction

Recognition of general, non-polyhedral 3D objects remains an active area of research in computer vision. Many feature-based systems have been developed and proven useful in the recognition of polyhedral objects. However, due to the nature of the features they utilize, namely points and lines, these systems are not suitable for recognizing generic 3D objects. Our philosophy is that to accomplish this task, it is necessary: to divide the general-object case into classes of objects; to utilize the appropriate sensors for each object class; and to make use of the appropriate features that can be reliably extracted using those sensors. We are currently working towards accomplishing this kind of generic recognition in scenes containing multiple 3D objects. The following work is most related or important to our own.

* This research was supported by the National Science Foundation under grant number IRI-9023977, by the Boeing Commercial Airplane Group, and by the Washington Technology Center.

The work conceptually closest to ours is that of Gremban and Ikeuchi [6]. They introduce a paradigm called appearance-based vision, in which a new step in the recognition process is introduced. This step predicts and analyzes the appearances of the object models based on the CAD data and on the physical sensor models. The prediction can be either analytical or based on synthesized images of the objects in the model database. The predicted appearance is the set of features that are visible under a specific set of viewing conditions. The analysis of the predicted appearance allows for the generation of an object recognition program, to be used in the on-line phase of the recognition process. This method is also known as VAC (Vision Algorithm Compiler) because it takes a set of object and sensor models and outputs an executable object recognition program. The framework is general in the sense that it does not require any specific type of sensor. Their system has successfully recognized simple objects from range data in a bin-picking environment. The major drawbacks of this approach are: 1) analytical prediction is impractical in some domains; and 2) synthetic images are not yet realistic enough for general use.

The PREMIO system of Camps et. al. [3] utilizes artificially rendered images to predict object appearances under various environmental conditions (sensor, lighting and viewpoint location). The predictions generated by the system did not agree well enough with the real images acquired under the same set of conditions. In order to improve PREMIO's predictions, Pulli [10] developed the TRIBORS system. He initially attempted to improve the predictions by using a better ray tracer, but that was also insufficient. The solution he found was to bootstrap the prediction process with synthetic images and to train on real images. These new predictions led to better and faster object recognition.

Despite the fact that it only deals with two-dimensional objects, Bolles and Cain's Local-Feature-Focus Method [2] is very relevant to our work. This method automatically analyzes the object models and selects the best features for recognition. Typical features include holes and corners. The basic principle is to locate one relatively reliable feature and use it to partially define a coordinate system within which a group of other key features is located. If enough of these secondary features are located and if they can uniquely identify the focus feature, then the hypothesized position and orientation of the object (of which this feature is a part) is determined. A verification step that utilizes template matching is then performed to prove or disprove the hypothesis. The system has been proven to efficiently recognize and locate a large class of partially visible two-dimensional objects.

The work of Murase and Nayar [9] also involves appearance of objects. They argue that since the appearance of an object is dependent on its shape, its reflectance properties, its pose in the scene, and the illumination conditions, the problem of recognizing objects from brightness images is more a problem of appearance matching than of shape matching. They define a compact representation of object appearance that is parametrized by pose and illumination only, since shape and reflectance are intrinsic (constant) properties. This represen-

tation is obtained by acquiring a large set of real images of the objects under different lighting and pose configurations, and then compressing the set into an eigenspace. A hypersurface in this space represents a particular object. At recognition time, the image of an object is projected onto a point in the eigenspace and the object is recognized based on the hypersurface on which it lies. The exact location of the point determines the pose of the object. The major drawback of this method is that it cannot handle multiple-object scenes. Occlusion also adversely affects the performance of the system.

Though the work of Bergevin and Levine [1] on generic object recognition does not make use of the specific model-based paradigm, it is related to ours in philosophy. They utilize coarse, qualitative models that represent classes of objects. Their work is based on the recognition by component (RBC) theory of Biederman. The system is divided into three main subsystems: part segmentation, part labeling, and object model matching. The part segmentation algorithm is boundary-based and it is independent of the specific shape of the parts making up an object. The part (geon) labeling algorithm makes use of the concept of faces to further categorize the geons into generalized solids. At the matching stage, the labeled geons are used to index into the database of models. A measure of similarity is defined in order to discriminate between the models. An important observation made by the authors themselves is that it is not clear that suitable line drawings may eventually be obtained from real images. All their examples and tests have made use of ideal line drawings.

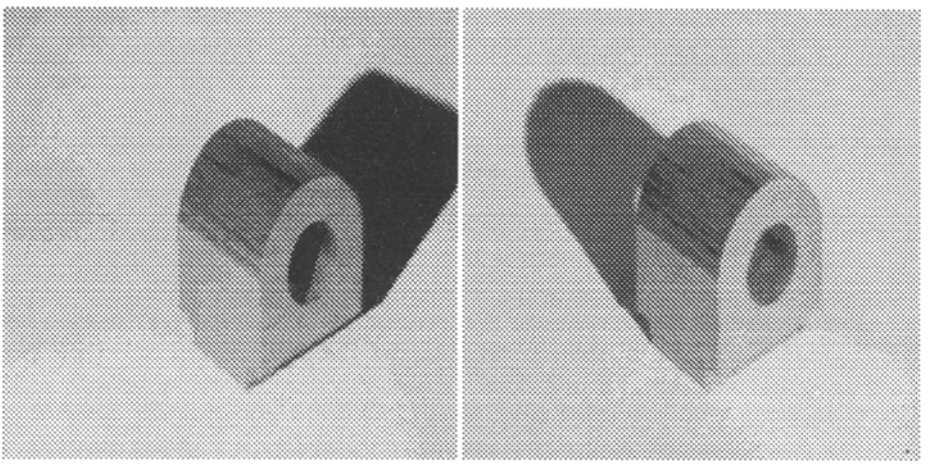
The evidence-based recognition technique proposed by Jain and Hoffman [8] defines an object representation and a recognition scheme based on salient features in range images. The objects are represented in terms of their surfaces, boundaries, and edges. The recognition scheme makes use of an evidence rulebase, which is a set of evidence conditions and their corresponding weights for various models in the database. The similarity between a set of observed image features and the set of evidence conditions for a given object determines whether there is enough evidence that the particular model is in the image. The model features must be carefully chosen in order to make possible the distinction between object classes.

2 Appearance-Based Models

The *appearance-based model* of an object is defined as the collection of the features that can be reliably detected from a training set of real images of the object. If a well-defined procedure exists through which a computer program can extract a given feature, this feature is said to be *detectable*. Even though appearance-based models can be *full-object models*, we choose to use *view-class models* in which an object is represented by a small set of characteristic views, each having its own distinct feature set [11]. Since we are currently only dealing with intensity images, all of our features are 2D features which may or may not directly correspond to a 3D feature, as in the case of limb edges.

Let $S_{V,M}$ be a set of training images for view class V of object model M. Each image $I \in S_{V,M}$ is processed to yield a set of features F_I. A feature f_n^I from image I is *equivalent* to another feature f_m^J from image J if they have the same type and are judged to have come from the same 3D source. The set of features that represent the view class is the set $F_{V,M}$ of equivalence classes of the union of the feature sets. The feature types we are investigating for use in our system are: coaxial circular arcs (two-cluster, three-cluster, and multi-cluster), ellipses, triples of line segments (U-shaped and Z-shaped), junctions (V-junction, T-junction, Y-junction, and Arrow), parallel line segments (close and far apart).

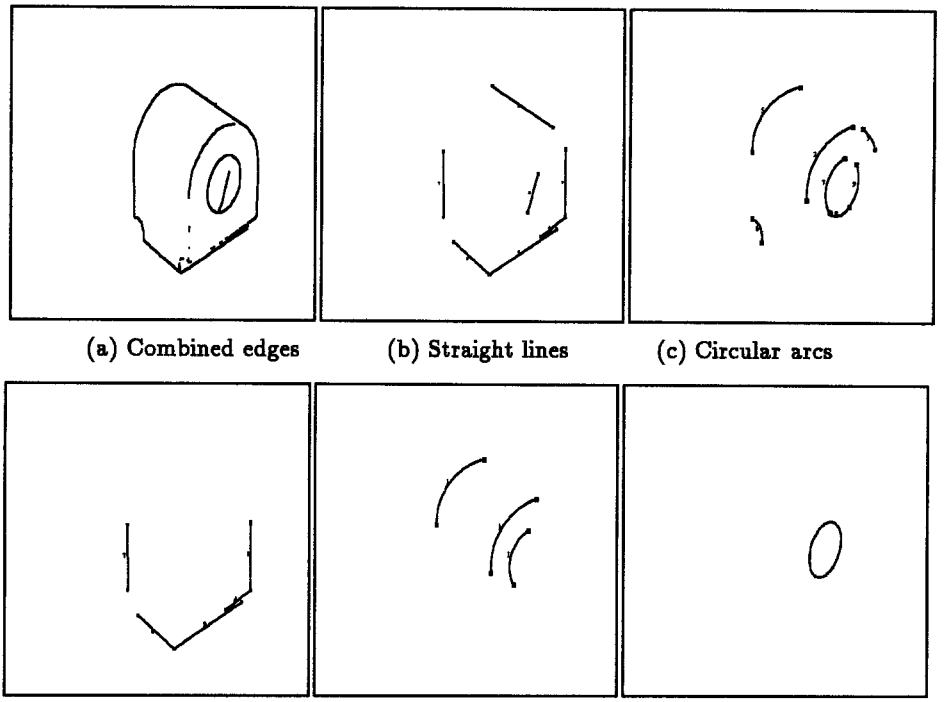
A natural extension to the use of features in a recognition task is the use of their properties and the relationships among them. In order to incorporate that, we define a view-class model by its *structural description* $D_{V,M} = (F_{V,M}, P_{V,M}, R_{V,M})$, where $P_{V,M}$ is a set of the properties of the features, and $R_{V,M}$ is a set of the relationships among the features.



(a) Light source at the left of camera (b) Light source at the right of camera

Fig. 1. Example of intensity image pair used by the system.

Our system works with pairs of intensity images. The two images are taken from the same viewpoint, but with two different lightings, one with the light source at the left and one with the light source at the right, as illustrated in Figure 1. By combining the two images, shadows can be eliminated and a more reliable edge image can be obtained. Figure 2 shows the features that were extracted by processing the sample image pair of Figure 1. Edges are detected using a Canny operator and the segmentation into lines and circular arcs is obtained using the Object Recognition Toolkit (ORT) package [5]. The line features (pair of parallel lines and the two V-junctions), the cluster of three circular arcs and the ellipse are detected by our system from the more primitive ORT features.



(a) Combined edges (b) Straight lines (c) Circular arcs

(d) Line features (e) Arc cluster (f) Ellipse

Fig. 2. A set of features extracted from the sample pair of Figure 1.

3 Scene Analysis Using Appearance-Based Models and Relational Indexing

We have created a database of appearance-based object models for a set of mechanical parts that have both flat and curved surfaces, holes, and threads. The structural descriptions $D_{V,M}$ of all the model-views were derived from a large set of training pairs of real images. We currently have 280 image pairs of 7 models. Our scene analysis paradigm makes use of the appearance-based models database and of a matching technique we call *relational indexing*.

The idea behind relational indexing is to utilize the structural description $D_{V,M}$ and represent each model-view as a relational graph $G_{V,M}$ of the features $F_{V,M}$ and relations $R_{V,M}$. The indexing principle is the same as in the original geometric hashing technique [7]. The method has two main phases: preprocessing and matching. The first one is an off-line phase in which the information contained in the entire database of models DB, is converted into a different representation that allows for a rapid retrieval. This is done in the following way: for each $G_{V,M}$ in DB, small relational subgraphs of size n are encoded and used as indices to access a hash table. The bin corresponding to a particular encoded subgraph stores information about which model-views gave rise to that particu-

lar index. This is done for all possible subgraphs of size n and for all the models in the database. Figure 3 shows a partial graph representing a view class of one of our objects (the "hexnut") and all the subgraph indices of size $n = 2$ for the given relational graph.

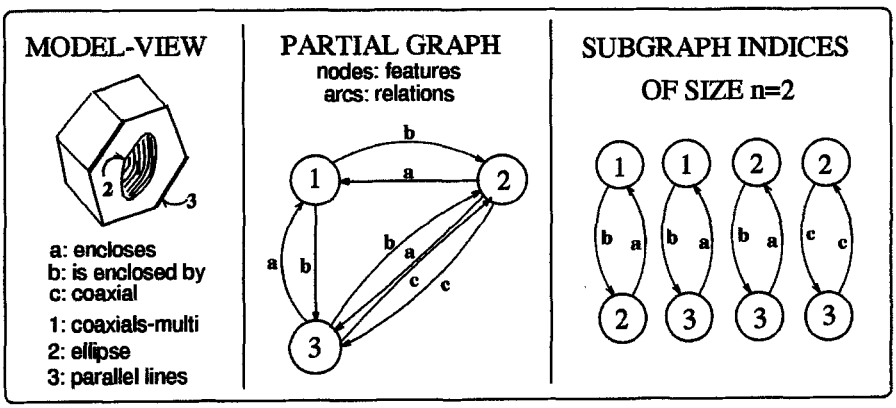


Fig. 3. Sample graph and corresponding subgraph indices of size 2.

In the second phase (performed at recognition time), a relational graph G_I of the input image pair is constructed using the features and relations detected in the scene. As in the off-line phase, all the subgraphs of size n are encoded and used to index into the hash table. Votes are cast for each model-view class stored in the bin indexed by each encoded subgraph. After all possible subgraphs have been used to index the table, the model-views with sufficiently high votes are taken as possible matching hypotheses. Details on the implementation of the hash table and the hashing scheme used can be found in [11].

Since some model-views share features and relations, it is expected that some of the hypotheses produced will be incorrect. This indicates that a subsequent verification phase is essential for the method to be successful. It is important to mention that the information stored in the hash table is actually more than just the identity of the model-view that gave rise to a particular subgraph index. It also contains information about which specific features (and their attributes) are part of the subgraph. This information is essential for hypothesis verification.

4 Multi-Level Indexing

In the case of single-object scenes, where there is no occlusion, one expects to extract most of the features and relations detected in the model-generation training phase. Therefore, the larger the subgraphs used, the more reliable and efficient the matching will be. However, in the case of multi-object scenes, only unoccluded objects will match to large subgraphs. Typically, in such scenes,

features are missing or are only partially detected, and may even become different features due to occlusion. Consequently, their original relations are also greatly affected. Thus, it is more appropriate to use relational subgraphs of small size (a low level of indexing), which will include only a couple of features and relations, since these are more immune to the adverse effects occlusion has on both the features and the relations.

Taking the above into consideration, it seems natural to consider a multi-level indexing approach to matching. Without any knowledge of the degree of occlusion in the scene, the system starts at the largest subgraph level and goes down to lower levels as necessary to recognize all objects in the scene. Objects that are unoccluded are expected to be recognized at the higher levels of indexing while highly occluded objects may only be recognized at the lowest levels.

5 Results and Discussion

In order to illustrate our scene analysis methodology, we matched the image of a scene containing four objects to the database of model-views. The database of models was created by encoding all relational subgraphs of size $n = 2$ for each of the model-views. The test image pair was processed, features and relations were detected, the relational graph was built, and all subgraphs of size $n = 2$ were encoded. The relational indexing was then performed and the generated hypotheses were normalized and ranked in order of strength. Among the five significantly high-ranked hypotheses, four were correct and they are shown in Figure 4. These hypothesized models were taken through pose computation (affine correspondence of appearance-based model features and scene features) without verification. The fifth strong hypothesis (not shown) matched the object "hexnut" to an incorrect view of the corresponding object model. The subgraph indices shown in Figure 3 are among those that were used in the matching process.

As it can be seen, the method produces promising results. A verification procedure is being designed in order to effectively rule out incorrect hypotheses that may be generated. Future work includes the development of a Bayesian approach to the relational indexing paradigm, along the same lines as our previous work on indexing [4], and the exploration of the proposed multi-level indexing technique applied to the case of scenes with a large degree of occlusion.

References

1. R. Bergevin and M. D. Levine. Generic Object Recognition: Building and Matching Coarse Descriptions from Line Drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(1):19–36, 1993.
2. R. C. Bolles and R. A. Cain. Recognizing and Locating Partially Visible Objects: The Local-Feature-Focus Method. *The International Journal of Robotics Research*, 1(3):57–82,1982.

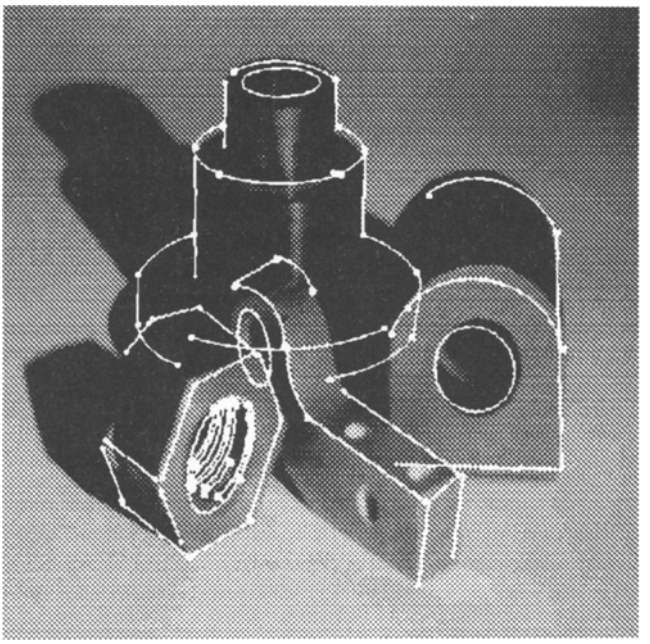


Fig. 4. Right image of a test scene overlaid with the appearance-based features of the hypothesized model matches.

3. O. I. Camps, L. G. Shapiro, and R. M. Haralick. Image Prediction for Computer Vision. In *Three-dimensional Object Recognition Systems*, A. Jain and P. Flynn (eds). Elsevier Science Publishers BV, 1993.
4. M. S. Costa, R. M. Haralick and L. G. Shapiro. Optimal Affine Invariant Point Matching. In *Proceedings of 10th ICPR*, volume 1, pp. 233–236, 1990.
5. A. Etemadi. Robust segmentation of edge data. In *Proceedings of the IEE Image Processing Conference*, 1992.
6. K. D. Gremban and K. Ikeuchi. Appearance-Based Vision and the Automatic Generation of Object Recognition Programs. In *Three-dimensional Object Recognition Systems*, A. Jain and P. Flynn (eds). Elsevier Science Publishers BV, 1993.
7. R. Hummel and H. Wolfson. Affine Invariant Matching. *DARPA Image Understanding Workshop*, April, 1988.
8. A. K. Jain and R. Hoffman. Evidence-Based Recognition of 3-D Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):783–802, 1988.
9. H. Murase and S. K. Nayar. Visual Learning of Object Models from Appearance. *International Journal of Computer Vision*, in press. Also Tech. Rep. CUCS-054-92.
10. K. Pulli. TRIBORS: A Triplet-Based Object Recognition System. *Technical Report 95-01-01*, Department of Computer Science and Engineering, University of Washington, January 1995.
11. L. G. Shapiro and M. S. Costa. Appearance-Based 3D Object Recognition. In *Proc. of the NSF/DARPA Workshop on 3D Object Representation for Computer Vision*, New York, NY, December 1994.