Probabilistic Hypothesis Generation for Rapid 3D Object Recognition

June-Ho Yi

School of Electrical and Computer Engineering
Sungkyunkwan University
Suwon 440-746, Korea
email: jhyi@ece.skku.ac.kr

Abstract. A major concern in practical vision systems is how to retrieve the best matched models without exploring all possible object matches. This research presents probabilistic hypothesis generation based on indexing approach for the rapid recognition of three dimensional objects. We have defined the discriminatory power of a feature for a model object is defined in terms of *a posteriori* probability. This measure displays belief that a model appears in the scene after a feature is observed. We compute off-line the discriminatory power of features for model objects from CAD model data using computer graphic techniques. In order to speed up the indexing or selection of correct objects, we generate and verify the object hypotheses for features detected in a scene in the order of the discriminatory power of these features for model objects. Experimental results on synthetic and real range images show the effectiveness of our probabilistic method for hypothesis generation.

Keywords: 3D, object recognition, probabilistic, indexing

1 Introduction

The fundamental issue in model-based recognition is how to rapidly narrow down the number of candidate models without actually searching through all the models. This problem has motivated the use of indexing or hashing for efficient retrieval of correct object model objects. In indexing, the feature correspondence and search of model database are replaced by a table look-up mechanism and this indexing table is computed off-line [1-2]. Recently, there have been some research works based on probabilistic indexing [3-4] where not only correspondence hypotheses but also the probability of each one being a correct interpretation is provided.

Wheeler and Ikeuchi [4] compiled statistical information about image features and object features off-line from a large set of ray-traced images of each object. They represented the likelihood of hypotheses and their inter-dependencies using MRF(Markov random field) to select a set of hypotheses with strong supporting evidence. Their system only considers polyhedral model and does not handle

C. Arcelli et al. (Eds.): IWVF4, LNCS 2059, pp. 630–639, 2001.

situations where only one surface is visible, while our system can handle objects with curved surfaces and single surface view situations. Beis and Lowe [3] also employed a probabilistic approach. They used 4-straight-line-segment chains (three angles and the ration of the interior edge lengths) as indexing vector and trained an indexing function (a linear combination of Gaussian centered on the indexing vectors) from synthetic images taken from various viewpoints. Their indexing vectors can not handle objects with curved edges. From the indexing function computed, they obtain the probability of each hypothesis being a correct interpretation of the data. Performance of these systems cannot be compared directly because they have been developed based on different assumptions and they perform in different scenarios using different features to generate object hypotheses.

We have employed a formal probabilistic solution for efficient indexing of correct model objects using a Bayesian framework. We define a decision-theoretic measure of the discriminatory power of a feature for a model object in terms of *a posteriori* probability. We estimate this measure off-line using computer graphic techniques. This measure allows us to employ salient features of model objects first for object recognition. In our system design, a measure of how well a feature can be detected, called "the detectability of a feature" is defined as a function of the feature itself, the viewpoint, sensor characteristics, and the feature detection algorithm. The detectability of a feature is incorporated into the formulation of the discriminatory power of a feature for a model object by considering model dependent information and sensing dependent information separately based on their conditional independence. In order to speed up the indexing or selection of the correct objects, we generate and verify the object hypotheses for the features detected in the scene, in the order of the discriminatory power of these features for model objects. By considering the object hypotheses in this order, we verify only a few correct hypotheses of the scene objects, resulting in the acceleration of recognition.

The following section gives a brief overview of our vision system. In section 3, we define a decision-theoretic measure of the discriminatory power of a feature for a model object. In section 4, we describe how object features for indexing are automatically compiled using our example feature, LSG (Local Surface Group). Section 5 presents experimental results on the effectiveness of our probabilistic indexing scheme.

2 System Overview

Let us briefly overview the entire object recognition system proposed. The system is divided into two parts. One is off-line compilation of model information and the other is on-line recognition.

The first component is concerned with the automatic computation of object representations from a CAD model database for recognition of model objects. The second component is the range image simulator. One module of this component is for simulating the sensing process to estimate the detectability of features.

The other module renders each model object for a viewpoint sampling. From the rendered images, knowledge of all objects in the domain of interest is compiled. After renderings are done for all model objects, the third component computes *a posteriori* probability that a model object appears in a scene given a detected feature, i.e., the discriminatory power of a feature for a model object. As a result, a feature indexing table is constructed where features are linked to the models with *a posteriori* probabilities. This indexing table is loaded at recognition time.

The on-line process consists of feature extraction, matching, and verification modules. The input to the feature extraction process is a dense range (depth) map from a single viewpoint. The feature extraction module detects features for generating object hypotheses. During the matching phase, features extracted from the scene are indexed by means of the precomputed indexing tables. A set of hypotheses are created and ordered in decreasing order of the probabilities associated with them. We validate these hypotheses applying a series of filters using geometric constrains. Finally, we obtain a list of valid hypotheses that will enter the verification stage. At the verification stage, the valid hypotheses are verified in the order they appear in the list (i.e., in the order of their probability).

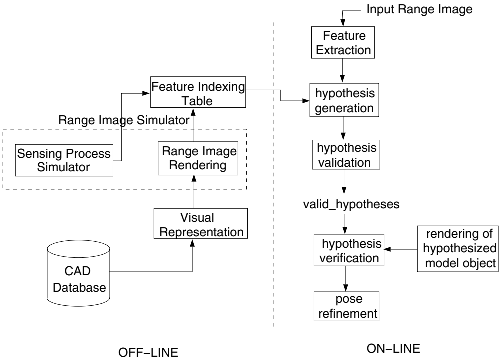


Fig. 1. System Overview

3 Discriminatory Power of a Feature for a Model Object

We exploit the discriminatory power of a feature for a particular model object for efficient indexing of the best matched models. In order to define the discriminatory power of a feature for a model object in terms of *a posteriori* probability, we start with the joint probability $P(m_k, M_i, viewpoint_j)$, $k = 1, \cdots, f$, $i = 1, \cdots, N$, and $j = 1, \cdots, v$ where m_k, M_i, and $viewpoint_j$ denote a feature for indexing, the i-th model object, and the j-th viewpoint of a set of viewpoint samplings, respectively. f, N, and v represent the numbers of features, models, and viewpoints, respectively. This joint probability encodes the information conveyed by a feature of a model object. The same feature may occur in several different models. If a feature to be used is viewpoint independent, we can ignore $viewpoint_j$ in $P(m_k, M_i, viewpoint_j)$.

3.1 Definitions and Notations

$P(m_k, M_i, viewpoint_j)$ **:** joint probability of m_k (a feature for indexing), M_i (i-th model object), and $viewpoint_j$ (j-th viewpoint of a set of viewpoint samplings), where $k = 1, \cdots, f$, $i = 1, \cdots, N$, and $j = 1, \cdots, v$.

$P(M_i)$ **:** The probability that a given object in a scene is M_i. Then we have $\sum_{i=1}^{N} P(M_i) = 1$.

$P(m_k/M_i)$ **:** a likelihood function, that is,
$P(m_k/M_i) > P(m_k/M_j)$ means that the model M_i is more "likely" to be the model object that the feature m_k belongs to than the model M_j, in that m_k would be a more plausible instance of the features of the model M_i than the model M_j.

$P(M_i/m_k)$ **:** This *a posteriori* probability reflects the updated belief that model M_i appears in the scene after the feature m_k is observed.

D_{m_k} **:** Detectability of a feature m_k. It measures how well a feature m_k can be detected.

Definition :
$P(M_i/m_k)$ is the discriminatory power of the feature m_k for a particular model object M_i.

The detectability of a feature is considered in the computation of the discriminatory power of a feature for a model object. The detectability of a feature, D_{m_k}, depends on the feature m_k itself (i.e. feature class). For example, a vertex feature may be less reliably detectable than a surface feature. D_{m_k} changes as the viewpoint varies. For example, when a planar surface is detected in various viewpoints, it is more difficult to detect in a viewpoint involving a very high sloped appearance of the planar surface than would be the case in a viewpoint giving a flat appearance of the planar surface. The sensor's capability is also important for a feature to be reliably detectable. Finally, the detectability of a feature can vary according to the feature detection algorithm used. Therefore, we represent the detectability of a feature m_k, D_{m_k} as a function of the above four factors:

$$0 \leq D_{m_k} = f(m_k,\ \text{viewpoint},\ \text{sensor},\ \text{feature detection algorithm}) \leq 1 \qquad (1)$$

3.2 Computation of Discriminatory Power

In the following, we will describe how to estimate *a posteriori* probability, $P(M_i/m_k)$. Let us denote estimates of quantities defined in the previous section by a hat above the symbol. $\hat{P}(M_i)$ and $\hat{P}(m_k, viewpoint_j/M_i)$ can be calculated once we know the specific application domain and determine which feature to use. $\hat{P}(M_i)$ can be computed by observing the frequency of the appearance of the model object M_i in the scene and normalizing it by the total number of observations of all models. Once the decision to use feature m_k for the indexing of model objects is made, we compute $\hat{P}(m_k, viewpoint_j/M_i)$ by counting the number of appearances of the feature m_k in model object M_i for the $viewpoint_j$. However, the feature m_k is often not perfectly detectable, i.e., D_{m_k} is not 1.0. To incorporate feature detectability into the computation of the discriminatory power, we consider model dependent information and sensing dependent information separately. That is, we estimate the model dependent information, $\hat{P}(m_k, viewpoint_j/M_i)$, assuming perfect detectability of the feature m_k and incorporate the sensing dependent information by multiplying these two terms as follows:

$$\begin{aligned}&\hat{P}(m_k, viewpoint_j/M_i) \cdot \hat{D}_{m_k}\\ &= \left(\frac{\#\text{ occurrences of the feature } m_k \text{ in } M_i \text{ for } viewpoint_j}{\#\text{ occurrences of all features } m_l, l=1,\cdots,f_m \text{ in } M_i \text{ for all viewpoints}}\right) \cdot \hat{D}_{m_k}\end{aligned} \tag{2}$$

This way, feature detectability can be incorporated into the computation of the discriminatory power when CAD model data is used. Therefore, the likelihood $\hat{P}(m_k/M_i)$ and *a posteriori* probability $\hat{P}(M_i/m_k)$ are computed as

$$\hat{P}(m_k/M_i) = \sum_{j=1}^{v} \hat{P}(m_k, viewpoint_j/M_i) \cdot \hat{D}_{m_k} \tag{3}$$

and

$$\hat{P}(M_i/m_k) = \frac{\hat{P}(M_i) \sum_{j=1}^{v} \hat{P}(m_k, viewpoint_j/M_i) \cdot \hat{D}_{m_k}}{\sum_{i=1}^{N} \hat{P}(M_i) \sum_{j=1}^{v} \hat{P}(m_k, viewpoint_j/M_i) \cdot \hat{D}_{m_k}}. \tag{4}$$

Note that if a feature m_k does not exist in the model object M_i, $\hat{P}(m_k/M_i) = 0.0$. As previously stated, the same formulation (3) and (4) can be applied to viewpoint independent features by ignoring the $viewpoint_j$ term.

Given a particular feature and viewpoint, estimating D_{m_k} amounts to determining how different feature detection algorithms behave under different sensor characteristics (for example, signal/noise ratio) [5]. For the case of edge detection in which the feature is an edge, the Sobel operator is known to perform better in noisy situation than the Robert's cross. Therefore D_{m_k} for edge features would be higher for the Sobel operator than for the Robert's cross. In fact, it is possible to analytically determine the probability of detecting an edge using either algorithm with a given signal/noise ratio. In the current prototype system, D_{m_k} is assumed a constant.

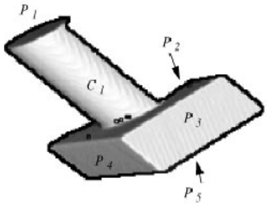
4 Construction of Indexing Table

In this section, we describe our example feature, LSG, for object hypothesis generation and present how to construct an indexing table.

4.1 Model Features for Object Hypothesis Generation

Our object recognition system can employ a wide class of features for object hypothesis generation as long as the discriminatory power, $P(M_i/m_k)$ can be computed. In the current prototype system, we use the LSG (local surface group). LSG is not a simple feature but a viewpoint dependent feature structure that contains several attributes. Figure 2 shows an example of a LSG that consists of a visible surface patch C_1 and its two adjacent surface patches, P_1 and P_2, that are simultaneously visible for the given viewpoint. Once we know the adjacent surfaces that are simultaneously visible, we access the node attribute set of the attribute-relational graph corresponding to the model object and can extract the information shown in the LSG. The most popular surface types used in computer vision are quadric surfaces because the majority of man-made objects can be modeled by them. Among the quadrics, *planar*, *cylindrical* (*ridge* and *valley*), and *spherical* (*peak* and *pit*) surfaces are supported in the current prototype system. The last entry of each surface patch in the attribute <simultaneously-visible-adjacent-surfaces: < list of surfaces >> is the angle between the surface orientation of the seed surface and the adjacent surface. This angle is only applicable only when two surface types are either planar or cylindrical (*ridge*, *valley*). Surface orientation is defined for planar surfaces as the direction of surface normal and for cylindrical surfaces as the direction of the axis, respectively. For pairs of other surface types, the angle value is set to NIL which indicates an undefined value. Note that the LSG is a viewpoint dependent feature structure and that the number of LSGs for a viewpoint is theoretically at most the number of visible surface patches for this given viewpoint. LSG can be extended to incorporate other feature attributes such as color and texture information.

Indexing involves a tradeoff between the complexity of the indexing feature and efficiency of indexing using the feature. If the indexing feature is complex (for example, a whole object as an indexing feature), indexing will not be efficient but will result in only a few candidate models. On the other hand, if a simple feature is used for indexing (for example, a surface patch as the indexing feature), indexing will be easy while many model objects are indexed for a feature detected in the scene. We will use a subset of the LSG as an indexing feature because the use of the complete LSG as an indexing feature makes the indexing of model objects complex and computationally expensive. Choosing an indexing feature of optimal complexity in the sense of recognition performance is a topic for further work. We will call our indexing feature "Indexing_LSG". An example of the Indexing_LSG that is used in our current system for the indexing of model objects is shown in Figure 2. In an Indexing_LSG, only the surface type information of simultaneously visible adjacent surface patches and the sum of the angles listed in the LSG are encoded without distinguishing respective adjacent surface patches. Different instances of this Indexing_LSG are the m_k's in section 3.



```
An example of LSG (Local Surface Group)
  {seed-surface-label: C 1
     < model-label: *** >
     < surface-type: ridge (1) >
     < radius: 0.75 >
     < simultaneously-visible-adjacent-surfaces: [
            ( P 1 , planar (0), area, radius (nil), 0 0 )
            ( P 2 , planar (0), area, radius (nil), 0 0 ) ] >
  }
Indexing LSG
     {surface-type-of-seed-surface: ridge (1)
        surface-types-of-
      < simultaneously-visible-: (planar (0), planar (0)) >
        adjacent-surfaces
      < sum-of-angles: 0 0 >
     }
```

Fig. 2. An example of LSG

4.2 Automatic Compilation of LSGs

A model object is taken from our model database, rendered using z buffer algorithm along with surface labels for a viewpoint sampling, and LSGs are compiled. To obtain a viewpoint sampling, we use the dual of a geodesic polyhedron with frequency Q (default 4) of geodesic division based on icosahedron. This dual polyhedron generates $10Q^2 + 2$ (default 162) viewpoints on a unit sphere. The process of compiling LSGs can be summarized as:

```
For $i = 1, 2, \cdots, N$
    For $j = 1, 2, \cdots, v$
        Render the range image of the object, $M_i$, for $viewpoint_j$
        along with surface labels.
        Scan the range image to collect LSGs.
    end for $j$
end for $i$
Compute $\hat{P}(M_i/m_k)$'s and return the indexing table.
```

5 Experimental Results

5.1 Extracting and Ordering Scene LSGs

To compute LSGs from an input scene image, we segment the image first and characterize feature attributes of surface patches such as primitive surface type, area/radius of surface, surface normal direction, and so on.

5.2 Probabilistic Hypothesis Generation

We have experimented our approach using the 20 object model database shown in Figure 3. We have visualized in Figure 4 the distribution of $\hat{P}(M_i/m_k)$. Let us make several comments about the information displayed in Figure 4. If the discriminatory power of a feature (m_k) for a model object (M_j) is 1.0 (i.e. $P(M_j/m_k) = 1.0$), the feature, m_k, is unique to the model object. In other words, if the feature, m_k, is detected in the scene, it is certain that the model M_j is in the scene. On the other hand, suppose that feature, m_2, is detected in the scene. Then, $\left[\hat{P}(M_3/m_2) = 0.425\right] > \left[\hat{P}(M_0/m_2) = 0.401\right] > \left[\hat{P}(M_5/m_2) = 0.174\right]$ indicates the belief that appearance of model objects in the scene is plausible in the order of M_3, M_0 and M_5. Model object, M_3, is hypothesized first and then M_0 and M_5. Similarly, when several features are detected in the scene, object hypotheses are generated in the order of the discriminatory power, $P(M_i/m_k)$.

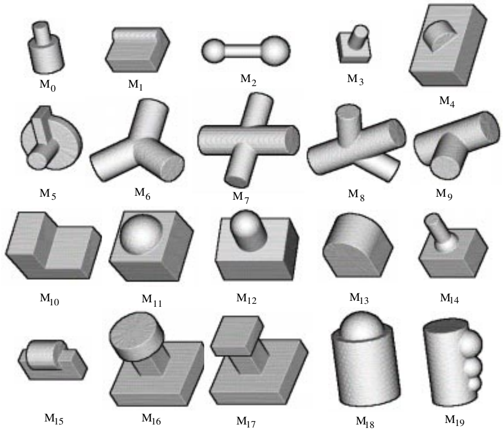


Fig. 3. Model database

5.3 Hypothesis Verification

We verify the hypotheses listed in the *valid_hypotheses*, one by one, in the order in which they appear. In order to verify an object hypothesis, we first find what model surfaces, other than the initially matched model surfaces in the hypothesis, should appear in the scene image. We compute a candidate view of the hypothesized model object from the initially matched pairs of scene and model surfaces in the object hypothesis. We render the hypothesized model object for the computed view and compute a list of neighboring surface pairs. Then the verification routine checks whether the model surface pairs can be found in the list of scene surface pairs. Compatibility between a model surface and the corresponding scene surface is determined based on the geometric constraints such as surface area, radius, and angle between two surfaces.

5.4 Indexing Efficiency

To experimentally determine the effectiveness of our indexing scheme, we define a measure of capability to index correct objects for our technique. We name this measure *indexing-efficiency-measure* and it is defined as:

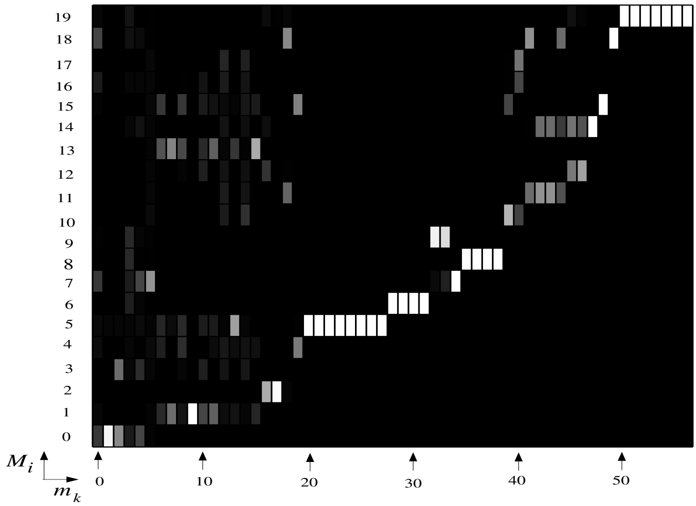


Fig. 4. Distribution of $P(M_i/m_k)$'s for 20 object model database shown in Figure 3. $P(M_i/m_k) = 0.0$ (black) and $P(M_i/m_k) = 1.0$ (white)

Definition:
indexing-efficiency-measure = position of the successfully verified hypothesis in the list of hypotheses initially generated.

We have experimented using a set of synthetic and real range images. A tabular summary of the results is shown in Table 1. We have generated synthetic range images of all objects in the model database for 10 randomly selected views for each object (total of 200 experiments). For real range images, we have built four objects, M_0, M_3, M_5 and M_{15}. 13 range images of these objects for several different poses were scanned. The average value of *indexing-efficiency-measure* was 2.80 and 2.68 for the synthetic and the real range images, respectively. That is, correct hypotheses were located near the third position in the list of hypotheses. This proves the effectiveness of our indexing scheme for the current model database although we adopted a simplified version of a LSG as an Indexing_LSGs. The average number of hypothesis verifications leading to successful recognition was 1.7 for the synthetic range images and 1.8 for the real range images, respectively, because hypothesis validation using geometric constraints served as an extra filter before each hypothesis entered actual verification procedure.

6 Summary and Conclusions

We have proposed a probabilistic method for efficient generation of object hypotheses, based on indexing approach. We achieve rapid recognition by generating the object hypotheses for the features detected in the scene in the order of the discriminatory power of these features for model objects. The discriminatory

Table 1. Experimental results

number of images	recognition accuracy	indexing-efficiency-measure
200 synthetic images	89.0% (178/200)	2.80
13 real images	84.6% (11/13)	2.68

power of an indexing-LSG in favor of an object model is computed off-line by compiling statistics from the rendered images of the model objects in the model database.

We experimentally proved the effectiveness of our indexing scheme using a feature structure called LSG (Local Surface Group) for generating the object hypotheses. The novelty of our approach is in the use of a formal probabilistic solution for efficient indexing of correct model objects, resulting in a speeding up of recognition.

Acknowledgements. This work was supported by grant number 1999-2-515-001-5 from the Basic Research program of the Korea Science and Engineering Foundation.

References

[1] Y. Lamdan, J. Schwartz, and H. Wolfson. Object recognition by affine invariant matching. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, June 1988.

[2] P. Flynn and A. K. Jain. Object recognition using invariant feature indexing of interpretation tables. *CVGIP: Image Understanding*, March 1992.
February 1992.

[3] J. Beis and D. Lowe. Learning indexing functions for 3-D model-based object recognition. In *AAAI Workshop*, April 1993.

[4] M. Wheeler and K. Ikeuchi. Sensor modeling, probabilistic hypothesis generation, and robust localization for object recognition. *IEEE Trans. Patt. Anal. Machine Intell.*, 17(3):252–265, March 1995.

[5] T. Kanungo, M. Y. Jaisimha, J. Palmer, and R. M. Haralick. A methodology for quantitative performance evaluation of detection algorithms. *IEEE Trans. Image Processing*, 4(12):1667–1674, December, 1995.