Towards A Robust Physics-Based Object Recognition System

Octavia I. Camps*

Dept. of Electrical Engineering
Dept. of Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802

Abstract. A successful 3D object recognition system must take into account imperfections in the input data, due for example to fragmentation or sensor noise. In this paper we propose a methodology for robust 3D object recognition using uncertain image data. In particular, we present a method capable of achieving acceptable performance in the presence of both segmentation problems and sensor uncertainty, thus eliminating the need for *ad hoc* heuristics. The proposed method is based upon the use of probabilistic models suggested by the underlying physics processes. These models are statistically validated and tested under controlled experimentation.

1 Introduction

Object recognition systems attempt to locate instances of objects in images. Most progress in this area has been made in industrial applications, such as robot manipulation and product inspection, where the visual environment can be controlled and the shape of the objects to be imaged is known in advance.

Many model-based systems find correspondences between model features and features detected in an image. Examples of features are points, edges, holes, junctions, or a combination of these. These correspondences are found using techniques such as interpretation trees [8, 6, 2], hashing [28, 4], alignment [14], and bipartite search [17]. The pairings are such that the features in the image can be obtained (approximately) by applying a geometric transformation to their corresponding model features. This transformation is usually referred as the *pose* of the object, that is the position of the object with respect to a coordinate system.

A successful 3D object recognition system must take into account imperfections in the input data, due for example to fragmentation or sensor noise. However, although there currently exists efficient model–based vision systems capable of recognizing and locating objects using nearly–perfect data, their performance degrades dramatically when confronted with real, non–perfect images. Recently some progress has been made in handling non-perfect data due to segmentation problems [1] and in analyzing the effect of sensor uncertainty [9].

* This work was supported in part by NSF grant IRI9309100 and in part by a Pennsylvania State University Research Initiation Grant.

In this paper we propose a methodology for robust 3D object recognition using uncertain image data. In particular, we present a method capable of achieving acceptable performance in the presence of both segmentation problems and sensor uncertainty, thus eliminating the need for *ad hoc* heuristics. The proposed method is based upon the use of probabilistic models suggested by the underlying physics processes. These models are statistically validated and tested under controlled experimentation.

2 Model Representation

The problem of describing the models is critical to the success of any object recognition system. *Characteristic views* [3, 15, 1] are commonly used to describe models for recognition purposes. A characteristic view is a representative view of a grouping of views or *view aspect* with similar properties.

The view aspect concept is very important in object recognition since it captures the topological characteristics of the views of an object. It allows a compact representation of the features of the models to be matched against the features in an image. Then, the object recognition/localization task can be divided into the following steps: (1) determine the correct view class; (2) find the correspondences between the features extracted from the image and those in the view class representation; and (3) use these correspondences and the links between the 3D features and the view class features to determine the pose of the object.

2.1 Characteristic Views

Characteristic views can be found by analytically partitioning a viewing sphere centered at the object into aspects [27, 5, 26]. The boundaries between these aspects are very accurate. However, the number of aspects tends to be large due to accidental viewpoints. An alternative approach, is to uniformly sample the viewing sphere around the object and to group together views that are "similar" [19]. This method results, in general, in a lesser number of aspects. However, the number of aspects will depend on the resolution of the sampling scheme and on the similarity measure used. In this paper, we use the later method, since it allows to limit the number of views and it is easy to implement.

The views obtained from the sampled viewing sphere are grouped into equivalence classes using a similarity metric. For this particular application we decided to cluster views depending on which model segments were observed in the views. Thus, each cluster had views in which roughly the same segments were observed. This is a simple but effective criteria for classification.

2.2 Probabilistic Prediction Models

In order to be robust to data uncertainty, physics-based knowledge of surface reflectance properties, light sources, sensor characteristics, and feature detector algorithms is incorporated to the view clusters.

For each view cluster C, we currently use the system PREMIO [1] to build a *probabilistic prediction model* combining hundreds of segmented views within the cluster. In the future, we will also use information obtained from real training images.

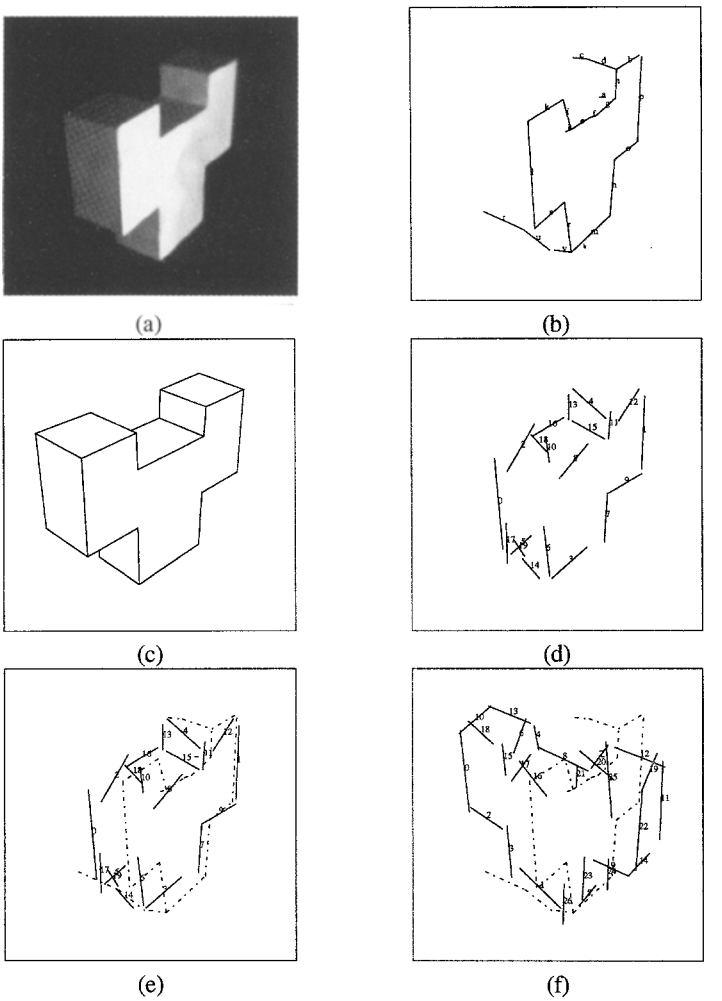

Fig. 1. (a) Fork image. (b) Segmented image. (c) Representative View. (d) Corresponding probabilistic model. (e) Alignment of (b) with (d). (f) Alignment of (b) with a different model.

A model M is represented by a quadruple $M = (L, R, f_L, g_R)$ where L is a set of model features or *labels*, R is a set of relational tuples of labels, f_L is the attribute value mapping that associates a value with each attribute of a label L, and g_R is the strength mapping that associates a strength with each relational tuple of R.

The set of labels L is formed by only those 2D features that have high enough probability of being detected for the given set of sensors and light sources. Furthermore, each feature in L has associated attributes which are given by the mean and the standard deviation of the attribute values of the feature for the n predictions. Similarly, the set of relational tuples R is formed for those relations among features in L such that they have high enough probability of holding. As with feature attributes, the relationship attribute

values of the tuples in R are represented by the mean and standard deviation of the relational tuples for the n predictions.

The model M obtained in this way, is a *probabilistic model* of the object for the given set of configurations of sensors and lights. Note that neither all the features in L, nor all the relational tuples in R need to be present in a single prediction. Neither do all the features of a particular prediction need to be in L. The model M combines a group of predictions into a single model, which is a sort of "average" model. The differences between the model M and the individual predictions that were used to build the model are summarized in a set of statistics Θ [1].

Figures 1(a) and (b) show an image of an object and the corresponding segmentation. Figure 1 (c) shows a representative view for the object in (a). Figure 1 (d) shows a visualization of the corresponding probabilistic model, where the segments are drawn using their mean attribute values and the numbers by the segments indicate their relative detectability, with the lower the number, the higher the detectability.

3 View Classification

Given an image, the objective is to find which object and in particular which view class it was originated from. Let $C_1, C_2, \ldots, C_n$ be a set of potential view clusters. Given an image I, our aim then, is to select the cluster C_i to which the image will most likely belong to. To achieve this, we use a Bayesian approach.

Let $P(C)$ be the *a priori* probability that an image from cluster C will be observed, and let $P(I|C)$ be the probability that a given image I is captured when the object is observed from a viewpoint within cluster C.

Then, given an image I, we will classify it as coming from cluster C_m, if the *a posteriori* probability $P(C|I)$ is maximum for $C = C_m$.

The *a posteriori* probability $P(C|I)$ can be computed using the Bayes' Theorem:

$$P(C \mid I) = \frac{P(I \mid C)P(C)}{\sum_{i=1,\ldots,n} P(I \mid C_i)P(C_i)} \tag{1}$$

3.1 Probabilistic Model

In order to apply (1), we need to compute the involved probabilities.

The probabilities $P(C)$ can be estimated from the area that the corresponding cluster spans on the viewing sphere. The larger the area, the higher the probability. The probabilities $P(I|C)$ depend on the selected features comprising the model. We will model this probability as a multivariate Gaussian distribution with mean vector $\underline{\mu}$ and covariance matrix Σ of the form:

$$P(I|C) = N_{\underline{x}}\left(\underline{\mu}, \underline{\Sigma}\right) \tag{2}$$

where $\underline{x}$ is $(d+4) \times 1$ feature vector representing the image I. The feature vector $\underline{x}$ consists of: the number of segments in the image, the number of junctions of 2 segments, the number of junctions of 3 segments, the number of triples[2], and a feature metric error

[2] A triple is an ordered set of three lines, with the lines traced clockwise.

for the d most detectable segments. The mean and covariance matrix of the distribution can be estimated from a set of samples generated with the prediction module of the PREMIO system.

Figure 1(e) shows the segmented image (b) aligned with the probabilistic model with the highest *a posteriori* probability, while (f) shows it aligned with a second model with lower probability.

4 Feature Selection

Once the object and view cluster are identified, the system proceeds to match a set of image and model features. The use of a small set of features is recurrent in the literature [22, 12, 23, 13, 10]. Since these correspondences will be used to compute the pose of the object, the problem of selecting "good" features is of interest. We propose to select these features based on measurements of their detectability, reliability, and accuracy, using concepts similar to the ones introduced by Ikeuchi and Kanade in [16].

4.1 Feature Detectability and Reliability

Let M be a model, L be a set of model features, I be an image of M, U be a set of image features, and $h : H \rightarrow U, H \subseteq L$ be the mapping that associates a set H of model features to the corresponding image features. Let l be a model feature, $l \in L$, and $u = h(l)$ be the corresponding image feature, $u \in U$. The feature *detectability* of l, denoted $D_M(l)$ is given by:

$$D_M(l) = P\left(l \in H \mid M\right) \quad . \tag{3}$$

Because a model can have several features with similar attributes, all of which could potentially be matched with the same image feature, feature detectability alone is not sufficient to determine which correspondence is correct. In order to determine the correctness of a match, the matching routine needs to know not only how detectable a model feature is, but also how "reliable" it is.

The feature *reliability* of l, denoted as $R_M(l)$ is defined as the probability that a hypothesized correspondence between a model and image feature will be correct given that the model feature is detected in the image:

$$R_M(l) = P\left(l \rightarrow h(l) \mid l \in H, M\right) \quad . \tag{4}$$

Feature reliability is an extension of feature detectability in the sense that a model feature must be detected in the image in order for a correspondence to be hypothesized and only then the correctness of the match can be assessed.

Computing feature detectability is rather straightforward. PREMIO approximates the detectability of a feature by the frequency at which it appears in a set of images. However, computing feature reliability is a more complex problem since it involves the process by which a correspondence is hypothesized. Next, we will discuss the theory supporting a method of computing feature reliability and illustrate its use in an object recognition system.

Let $\mathcal{I}$ be the set of all possible images of M, then $R_M(l)$ can be expressed as the feature reliability contributions for l integrated (summed) over all images of $\mathcal{I}$. Formally,

$$R_M(l) = \int_{\mathcal{I}} P\left(l \rightarrow h(l) \mid I, l \in H, M\right) . P\left(I \mid l \in H, M\right) dI \tag{5}$$

The reliability depends on how correspondence hypotheses are made. We assume that the matching strategy will hypothesize a correspondence between a label and a unit only if they are "sufficiently similar".

Let $\rho(l, u)$ be a metric that measures the similarity between the feature attributes of l and $h(l)$, where $h(l)$ is the observation of l in an image. Based on our experiments and the Central Limit Theorem, the similarity between the label l and its corresponding unit can be modeled as a Gaussian distribution, denoted $P_{\rho_l}(\rho_l) = N_{\rho_l}(\mu_{\rho_l}, \sigma_{\rho_l})$.

Given $P_{\rho_l}(\rho_l)$, it is natural to hypothesize a match between l and u only if

$$||\rho(l, u) - \mu_{\rho_l}||^2_{\sigma_{\rho_l}} \leq Th \tag{6}$$

where $||\rho(l, u) - \mu_{\rho_l}||^2_{\sigma_{\rho_l}}$ is the squared Mahalonabis distance between $\rho(l, u)$ and the expected value of $\rho(l, h(l))$ and Th is a threshold measuring the maximum allowed difference between the observed metric value and its expected value.

Consider the set of image features, U, and a label $l \in L$. Then, the subset of units that could potentially be matched with l is the subset of units, C, such that they satisfy (6):

$$C = \left\{ u \;\middle|\; u \in U, ||\rho(l, u) - \mu_{\rho_l}||^2_{\sigma_{\rho_l}} \leq Th \right\} .$$

Thus, the first factor in the right side of (5) can be expressed as

$$P\left(l \rightarrow h(l) \mid I, l \in H, M\right) = \begin{cases} 0 & \text{if } h(l) \notin C \\ \frac{1}{\#C} & \text{otherwise} \end{cases} \tag{7}$$

where # denotes cardinality.

The second factor in the right side of (5) can be expressed using Bayes rule and (3) as:

$$P\left(I \mid l \in H, M\right) = \frac{P\left(l \in H \mid I, M\right) P\left(I \mid M\right)}{D_M(l)} .$$

The probability $P\left(l \in H \mid I, M\right)$ is the probability that label l is detected given that the observed image is I and the model is M, and it can be modeled as

$$P\left(l \in H \mid I, M\right) = \begin{cases} 0 \text{ if } \#C = 0 \\ 1 \text{ otherwise} \end{cases} .$$

Finally, $P\left(I \mid M\right)$ is provided by PREMIO's statistics θ [2].

RD Heuristic

Feature reliability and detectability can be combined to form a joint ("RD") heuristic in order to produce a matching routine which is more efficient than a routine using detectability alone. The RD joint heuristic can be stated as the joint probability of feature reliability $R_M(l)$ and detectability $D_M(l)$ denoted as $P\left(l \rightarrow h(l), l \in H \mid M\right)$. By definition of conditional probability and (3) and (4):

$$\begin{aligned} & P(l \rightarrow h(l), l \in H|M) = \\ & = P\left(l \rightarrow h(l) \mid l \in H, M\right) P\left(l \in H \mid M\right) = \\ & = R_M(l) D_M(l) \, . \end{aligned}$$

The probability $P\left(l \rightarrow h(l), l \in H \mid M\right)$ can then be used to rank the labels to be matched. Thus, a matching routine can use these rankings to make more reliable hypotheses reducing potential mistakes and expensive backtracking.

Figure 2 (a) shows the new rankings for the model given in Figure 1 (d) when the RD heuristic is used. Figures 2 (b) to (d) compare the performance in terms of their CPU time, probability of false alarm (finding the wrong correspondences) and probability of misdetection (not finding a set of correspondences) of an iterative deepening search algorithm [2] when only detectability or both detectability and reliability heuristics are used. It is observed that both CPU time and false alarms are always less for the RD heuristic. The probability of misdetection is also better for the RD heuristic up to 13 correspondences.

4.2 Feature Accuracy

Most methods to compute the pose use a *few* point-to-point [11, 8] or line-to-line [20, 21] correspondences. If the data is perfect with no sensor uncertainty and with no incorrect correspondences, then the pose is exact, and the transformed model features exactly coincide with the image features. However, in most real cases the noise in the data will propagate into the pose. Moreover, the extent of the effect of the uncertainty depends on the correspondences used to compute it.

Currently, we are investigating the following problem:

> Let N be the number of model features and $n \leq N$ be the (small) number of features that will be assigned a correspondence and will be actually used to compute the pose of the object. Then, find the subset of n model features such that the effect of the data uncertainty in the estimation of the pose is minimized.

In [7] we reported a suboptimal solution for this problem for the special case when the features are points, the pose estimation algorithm is an iterative least square procedure, and the translation of the camera is constrained. In this case, the sensitivity of the pose estimation algorithm to the noise in the data is given by the amount of perturbation on the rotation matrix due to a small perturbation in the data. Using a sensitivity analysis similar to the one presented in [11] we showed that the trace of the matrix $(J'J)^{-1}$, where J is the Jacobian matrix of the image points with respect to the incremental correction

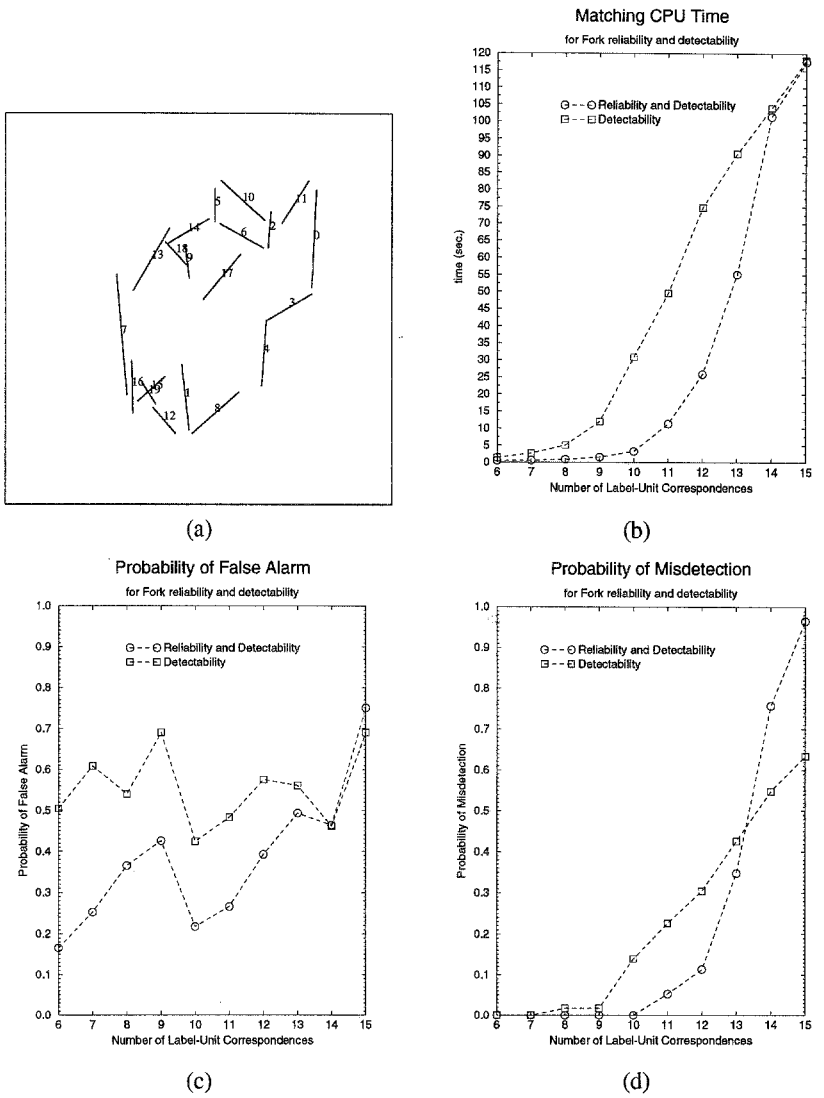

Fig. 2. (a) RD segment ranking. (b) Matching CPU time. (c) Probability of False Alarm. (d) Probability of Misdetection.

on the pose, can be used as a measure of the sensitivity of the pose to the noise in the data. Thus, the subset of n points that minimizes the trace of the above matrix:

$$\min_{\text{Subsets of } n \text{ points}} \left\{ \text{trace } (J'J)^{-1} \right\} \tag{8}$$

is a good choice from these considerations. Unfortunately, the number of possible subsets of n points is, in general, too large to attempt to solve this minimization problem

directly. However, a suboptimal solution to this problem can be obtained by using an incremental approach:

The Greedy Algorithm

> Given a subset $\mathcal{P}_k$ of k model points, select a model point m such that the trace of $(J'J)^{-1}$ is minimized for the extended subset $\mathcal{P}_{k+1} = \mathcal{P}_k \cup \{m\}$.

The proposed algorithm is a *greedy* algorithm that finds a model point such that when it is added to the points used so far, the computed pose is robust to the noise present in the data. The algorithm is suboptimal since it has a limited horizon of one point at a time.

Selection of initial points. The pose estimation algorithm starts with an initial rotation $R^{(0)}$ and then iterates to refine this pose. In order to compute the initial rotation $R^{(0)}$, a minimum of two points correspondences are required. Let m_1 and m_2 be two model points. It can be easily shown that if m_1, m_2, and the origin of the world reference frame are aligned, i.e. $m_2 = \alpha.m_1$, the estimated pose is not unique. Furthermore, if the points m_1 and m_2 are close to the origin, a small perturbation in the coordinates of the corresponding image points leads to a large change in the estimated pose.This suggests the heuristic rule that the initial points should be selected such that the area of the triangle formed by the two model points and the origin is maximum.

Selection of subsequent points. Once k correspondences have been found the problem of selecting the next correspondence such that the estimated pose is robust reduces to selecting the model point that minimizes the trace of the 3×3 matrix $(J'J)^{-1}$ that can be computed incrementally [7].

Handling outliers. The initial rotation $R^{(0)}$ is found by solving a system with four equations and three unknowns such that a least square error criterion is minimized for the two initial model points. If the error of this fit is too high, at least one of the points is likely to be an outlier and a newpair of points is selected.

When a subsequent point is added, one can use the current rotation $R^{(k)}$ to project the model points currently used and compare their location with their corresponding image points. If the distance between these is higher than a multiple of the standard deviation of the noise, then the point is rejected as an outlier.

If at a given point, too many points are classified as outliers, the initial points are suspected as outliers and the process starts again for a new pair of initial points.

Fig. 3 shows the results obtained with an image of a bookend. Fig. 3(a) shows a grayscale image of the bookend with the model points highlighted. Fig. 3(b) shows the back projection of the model when all the model points are used to compute the pose. Fig. 3(c) shows the back projection of the model onto the image when four random points (circled on the figure) are used to compute the pose. Finally, Fig. 3(d) shows the back projection of the model when four points are selected using the greedy algorithm (circled on the figure). Clearly, the greedy solution is better than the random one, and comparable to the one obtained using all the points.

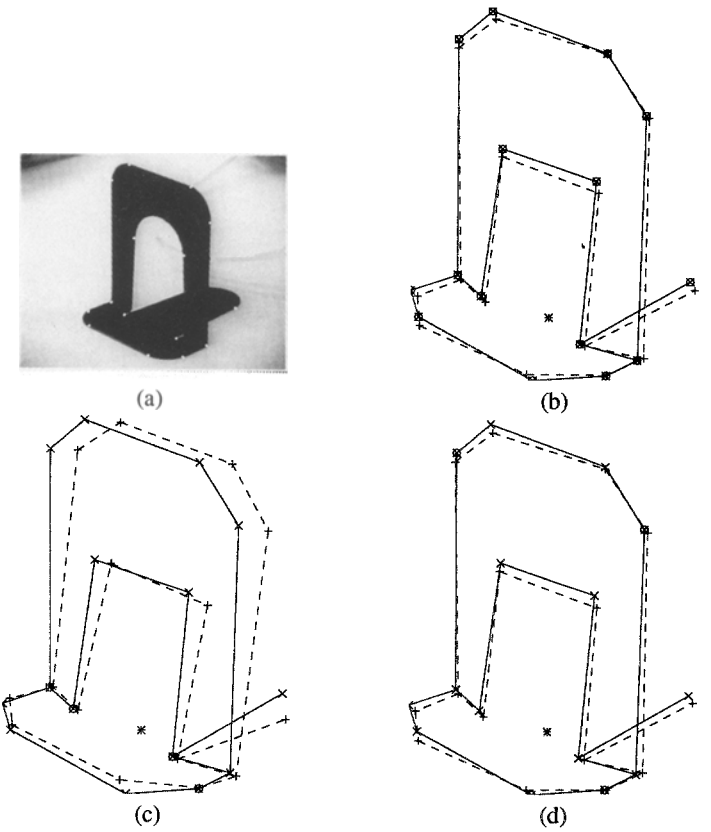

Fig. 3. Bookend image. (a) Grayscale image with model points highlighted. (b) Back projection using all the model points. (c) Back projection using four random model points (circled on the figure). (d) Back projection using four points selected using the greedy algorithm (circled on the figure).

We are currently working on the generalization of this algorithm to the case when the features are line segments. We plan to combine the new results with our current detectability and reliability heuristic to form a joint heuristic to guide the matching routine in the selection of the next correspondence to be sought.

5 Performance Evaluation

The importance of controlled experiments has only recently been stressed in computer vision. Controlled experiments are essential in order to illustrate the validity of a solution presented. We tested all the modules described here using artificially generated data as well as real images. The experimental protocol used has six steps: 1) modeling of the ideal inputs and the random perturbation processes; 2) annotating ground truth data; 3) estimating the free parameters of the random models; 4) statistically validating these models; 5) testing the algorithms; and 6) analyzing the results.

Appendix A gives the experimental protocol used to evaluate the performance of the RD heuristic. A detailed description of the protocol for the view classification module is given in [25], and for the greedy algorithm to select features based on pose accuracy is given in [7].

6 Discussion

A successful 3D object recognition system must take into account imperfections in the input data, due for example to fragmentation or sensor noise. However, although there currently exists efficient model–based vision systems capable of recognizing and locating objects using nearly–perfect data, their performance degrades dramatically when confronted with non–perfect images. We believe that to overcome these problems we must develop a *robust* 3D object recognition paradigm. Specifically, we need to:

1. Develop mathematical models for robust 3D object recognition from uncertain 2D image data.
2. Develop matching schemes that use these models to robustly recognize and compute the pose of an object. These schemes should also provide levels of confidence for the hypothesis made.
3. Develop thorough experimental protocols to characterize the performance and robustness of the systems.

We believe that techniques from robust statistics coupled with physics-based knowledge in a Bayesian framework are promising tools to achieve these goals. Our preliminary results show that they naturally lead to rigorous models capturing the underlying physical processes and that they are subject to experimental validation.

References

1. O. I. Camps, L. G. Shapiro, and R. M. Haralick. Image prediction for computer vision. In Jain A.K. and P.J. Flynn, editors, *Three-dimensional Object Recogntion Systems*. Elsevier Science Publishers BV, 1993.
2. O. I. Camps, L. G. Shapiro, and R. M. Haralick. A probabilistic matching algorithm for computer vision. *Annals of Mathematics and Artificial Intelligence*, 10, 1994.
3. I. Chakravarty and H. Freeman. Characteristic views as a basis for three-dimensional object recognition. In *SPIE 336 (Robot Vision)*, pages 37–45, 1982.
4. M. Costa, R.M. Haralick, and L.G. Shapiro. Optimal affine-invariant point matching. In *Proc. of the International Conference on Pattern Recognition*, pages 233–236, Atlantic City, New Jersey, June 1990.
5. D. Eggert. *Aspect Graphs of Solids of Revolution*. PhD thesis, Department of Computer Science and Engineering, University of South Florida, Tampa, Florida, 1991.
6. P.J. Flynn. *CAD-Based Computer Vision: Modeling and Recognition Strategies*. PhD thesis, Michigan State University, 1990.
7. T. L. Gandhi and O. I. Camps. Robust feature selection for object recognition using uncertain 2D image data. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 281–287, June 1994.

8. W.E.L. Grimson. *Object Recognition by Computer: The role of geometric constraints*. MIT Press, Cambridge, MA, 1990.
9. W.E.L. Grimson. Recognizing 3D Objects from 2D Images: An Error Analysis. Technical Report A.I. Memo No. 1362, MIT, Cambridge,MA, November 1992.
10. Charles D. Hansen. *CAGD-Based Computer Vision: The Automatic Generation of Recognition Strategies*. PhD thesis, The University of Utah, 1988.
11. R.M. Haralick and L.G. Shapiro. *Computer and Robot Vision*. Addison-Wesley, 1992.
12. J. Henikoff and L. Shapiro. Interesting patterns for model–based matching. In *ICCV*, 1990.
13. P. Horaud and R.C. Bolles. 3DPO: A system for matching 3-D objects in range data. In A.P. Pentland, editor, *From Pixels to Predicates*, pages 359–370. Ablex Publishing Corporation, Norwood, New Jersey, 1986.
14. D.P. Huttenlocher and S. Ullman. Object recognition using alignment. In *Proceedings of the First International Conference on Computer Vision*, pages 102–111, 1987.
15. K. Ikeuchi. Generating an interpretation tree from a CAD model for 3D-Object recognition in bin-picking tasks. *Int. J. Comp. Vision*, 1(2):145–165, 1987.
16. K. Ikeuchi and T. Kanade. Modelling sensor detectability and reliability in the configuration space for model-based vision. Technical Report CMU-CS-87-144, Carnegie-Mellon University, Computer Science Department, July 1987.
17. W.Y. Kim and A.C. Kak. 3D Object Recogntion Using Bipartite Matching Embedded in Discrete Relaxation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(3):224–251, March 1991.
18. D.E. Knuth. *The Art of Computer Programming, vol. 2*. Addison-Wesley, 1969.
19. M. Korn and C. Dyer. 3D-Multiview Object Representations for Model–Based Object Recognition. *Pattern Recognition*, 20(1):91–103, 1987.
20. R. Kumar and R. Hanson. Analysis of different robust methods for pose estimation. In *Proc. of IEEE Workshop on Robust Computer Vision*, October 1990.
21. C.N. Lee and R.M. Haralick. Exterior orientation from line-to-line correspondences – a Bayesian approach. In *Proc. of the IEEE Computer Vision and Pattern Recogntion*, June 1993.
22. D. G. Lowe. Three–dimensional object recognition from single two–dimensional images. *Artificial Intelligence*, 31:355–395, 1987.
23. R. Mohan and R. Nevatia. Using perceptual organization to extract 3d structures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(11):1121 – 1139, November 1989.
24. A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, third edition, 1991.
25. A. Pathak and O. I. Camps. Bayesian view class determination. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 1993.
26. J. Ponce and D. Kriegman. Computing exact aspect graphs or curved objects: parametric surfaces. In *8th National Conference on AI*, pages 340–350, 1987.
27. J.H. Stewman. *Viewer Centered Representations for Polyhedral Objects*. PhD thesis, Department of Computer Science and Engineering, University of South Florida, Tampa, Florida, 1991.
28. H.J. Wolfson. Model based object recognition by geometric hashing. In *First Europ. Conf. on Comp. Vis.*, pages 526–536, 1990.

A RD Heuristic Experimental Protocol

A.1 Probabilistic Prediction Model Generation

Model Generation. PREMIO generates the model M using the following steps:

1. Select a region $\mathcal{V}$ of the viewing space. The region $\mathcal{V}$ is a spherical sector between two spheres. It is specified by the range of the longitude $(\Phi_{v\min}, \Phi_{v\max})$ and latitude $(\theta_{v\min}, \theta_{v\max})$ angles and the radius of the viewing sphere $(R_{v\min}, R_{v\max})$.
2. Select a region $\mathcal{I}$ of the illumination space. This region is specified by the range of the longitude $(\Phi_{i\min}, \Phi_{i\max})$ and latitude $(\theta_{i\min}, \theta_{i\max})$ angles and the radius of the illumination sphere $(R_{i\min}, R_{i\max})$.
3. Given the desired number of samples N_v and N_i, sample the viewing and illumination space regions. Let $\mathcal{V}_s$ be the set of the sampled viewing positions and $\mathcal{I}_s$ the set of the sampled lighting positions.
4. Generate the predictions. For each pair $(v, i) \in \mathcal{V}_s \times \mathcal{I}_s$, predict the subset of detectable labels L_{vi}, its associated attribute mapping $f_{L_{vi}}$, the subset of detectable relational tuples R_{vi}, and its associated relationship strength mapping $g_{R_{vi}}$. Also generate the corresponding set of units U_{vi}, the associated attribute mapping $f_{U_{vi}}$, the set of relational tuples S_{vi}, and the associated strength mapping $g_{S_{vi}}$.
5. Obtain the detectability values for each label l. The previous step produced $N_v \times N_i$ different predictions. Approximate the probability of a label/relationship being detected, given that the view and light are in the specified regions $\mathcal{V}$ and $\mathcal{I}$, by the observed frequency rate of their detectability in the generated predictions. These approximations are based on the fact that the predictions were made from uniformly sampled camera and light positions as well as on the central limit theorem assuming that N_v and N_i are large enough.
6. Select the desired minimum label detectability t_f and the minimum relational detectability t_r.
7. Combine the $N_v \times N_i$ predictions into a single model $M = (L, R, f_L, g_R)$ such that the labels in L have a detectability greater than t_f and the relational tuples in R have a detectability greater than t_R.
8. Compute the *similarity probability distribution* $P_\rho(l)$ for each label in L.
9. Obtain the reliability values for each label in L.
10. Compute the joint reliability and detectability values for each label in L.

Statistics Generation. PREMIO generates the statistics Θ as follows:

1. For each generated prediction, PREMIO finds the true observation mapping between the predicted image and the obtained model M, $h_{vi} : H_{vi} \rightarrow U_{vi}$, with $H_{vi} \subseteq L$, and $(v, i) \in \mathcal{V}_s \times \mathcal{I}_s$. These observation mappings only include correspondences with units that were originated from labels with detectability greater than or equal to t_f.
2. Obtain matching errors. The previous step produced $N_v \times N_i$ true observation mappings, h_{vi}. For each prediction PREMIO computes the quantities: $\#L + \#U_{vi} - 2\#H_{vi}$, $\#(R - S_{vi} \circ h_{vi}^{-1}) + \#(S_{vi} - R \circ h_{vi})$, $\rho(f_{U_{vi}} \circ h_{vi}, f_{L|H_{vi}})$, and $\rho(g_{S_{vi}} \circ h_{vi}, g_{R|H_{vi}})$.
3. PREMIO uses the matching errors generated in the previous step to estimate the parameters of the four Gaussian distributions P_U, P_S, P_{f_U}, and P_{g_S}: μ_f, σ_f, μ_R, σ_R, μ_{f_U}, σ_{f_U}, μ_{g_S}, and σ_{g_S} by using the sample means and variances.

Model Validation. The PPM model obtained must be validated using statistical tests such as the Kolgomorov Smirnov test and the Chi-square test [24, 18]. If the model does not pass the tests, it can be rectified by one or more of the following methods:

1. Increase the number of predictions by increasing N_v and/or N_i. By increasing the number of predictions, the confidence interval of the statistics Θ, is narrowed, and hence a better estimation of the model is obtained.
2. Reduce the extension of the viewing region $\mathcal{V}$ and/or the illumination region $\mathcal{I}$. The failure to pass the test may be due to large dissimilarities among the views used to generate the model. By reducing the extensions of the regions $\mathcal{V}$ and $\mathcal{I}$, we can increase the similarity between these views.
3. Try different probability distributions, such as truncated Gaussian or double exponential distributions. The error distributions are defined only for *positive* errors. Hence, the approximation of an error distribution to a Gaussian distribution is only valid if its mean is more than two standard deviations away from the origin. If that is not the case, an asymmetric distribution, such as a truncated Gaussian or a truncated double exponential distribution, should be used.

A.2 Image Generation

PREMIO's matching routine was tested on simulated images using the joint feature reliability and feature detectability of the PPM labels as a heuristic to determine which labels should be matched first. PREMIO generates simulated images using the following steps:

1. Given the desired number of samples N_v' and N_i', PREMIO uniformly samples the viewing and illumination space regions $\mathcal{V}$ and $\mathcal{I}$. Let $\mathcal{V}_s'$ be the new set of the sampled viewing positions and $\mathcal{I}_s'$ the new set of the sampled lighting positions.
2. Generate the images. For each pair $(v, i) \in \mathcal{V}_s' \times \mathcal{I}_s'$, PREMIO predicts the subset of detectable labels L_{vi}, its associated attribute mapping $f_{L_{vi}}$, the subset of detectable relational tuples R_{vi}, and its associated relationship strength mapping $g_{R_{vi}}$. PREMIO also generates the corresponding set of units U_{vi}, the associated attribute mapping $f_{U_{vi}}$, the set of relational tuples S_{vi}, and the associated strength mapping $g_{S_{vi}}$.

A.3 Matching Routine

The effect of the joint feature reliability and feature detectability values assigned to the PPM labels was tested using PREMIO's matching routine. PREMIO attempts to match the randomly-generated images against the PPM using the joint feature reliability and feature detectability values as a heuristic to determine which labels the matching routine should attempt to match first. The matching algorithm was run several times with each successive run attempting to match more correspondences. The performance of the matching routine was evaluated by generating a *receiver operating curve* plot in which the probability of *misdetection error* was plotted against the probability of *false alarm error* over the range of correspondences sought. The following steps summarize PREMIO's matching routine:

1. Choose the number n of correspondences to match.
2. For each of the generated images:
   (a) Run the matching routine to search for an observation mapping having n correspondences between the PPM and the image.
   (b) If the observation mapping is found in a reasonable amount of time, then calculate the number of incorrect correspondences present in the mapping.
   (c) If the number of incorrect correspondences was calculated, then calculate the ratio of the number of incorrect correspondences to n for the given observation mapping. This ratio, denoted f, is the *false alarm ratio* and is defined as:

$$f = \frac{\#\text{Incorrect Correspondences}}{n} . \tag{9}$$

   The ratio f is inversely proportional to the strength of the mapping found.

A.4 Evaluation of Matching Results

If PREMIO does not find an observation mapping for the given number of n correspondences, then the experiment is termed a misdetection error (ME). The probability of a misdetection error, given n, is defined as:

$$P(\text{ME} \mid n) = \frac{\#\text{Mappings not Found}}{\#Images} . \tag{10}$$

If PREMIO finds an observation mapping, then the correctness of the observation mapping is determined by comparing the false alarm ratio f to the false alarm threshold ratio F. If f is larger than F, then the observation mapping is incorrect and the experiment is termed a false alarm error (FAE). F was varied in order to test the performance of the system. The probability of a false alarm error, given that a mapping m_n with n correspondences was found and given F, is defined as:

$$P(\text{FAE} \mid m_n, F) = \frac{\#\text{Incorrect Mappings}}{\#\text{Mappings Found}} \tag{11}$$

Equations (10) and (11) measure the performance of the matching routine. The performance of the matching routine for various values of n and F can be shown on receiver operating curves (ROC) where probability of misdetection is plotted against probability of false alarm error.