Bayesian View Class Determination *

Anjali Pathak
email: anjali@.whale.ece.psu.edu
Dept. of Electrical and Computer Engineering
The Pennsylvania State University
University Park, PA 16802

Octavia I. Camps
email: camps@whale.ece.psu.edu
Dept. of Electrical and Computer Engineering
The Pennsylvania State University
University Park, PA 16802

Abstract

To recognize objects and to determine their poses in a scene we need to find correspondences between the features extracted from the image and those of the object models. Models are commonly represented by describing a few characteristic views of the object representing groups of views with similar properties.

Most feature-based matching schemes assume that all the features that are potentially visible in a view will appear with equal probability, and the resulting matching algorithms have to allow for "errors" without really understanding what they mean. PREMIO is an object recognition system that uses CAD models of 3D objects and knowledge of surface reflectance properties, light sources, sensor characteristics, and feature detector algorithms to generate probabilistic models for a given view cluster.

The purpose of this paper is to present a Bayesian approach to the problem of given an image, how to determine the correct view class it belongs to, using the probabilistic models produced by PREMIO.

1 Introduction

Model-based vision systems are very useful for industrial vision tasks where CAD models of the parts to be manipulated or inspected are already available. Other areas where vision systems based in CAD models are useful are applications in hazardous environments, such as nuclear plants and nuclear waste management. Examples of such vision systems are found in [1, 4, 6, 9, 11, 13, 14].

A model based vision system attempts to find correspondences between features of a model object and features detected in an image for purposes of recognition, localization or inspection. Critical to this is the problem of describing the models. A common approach is to describe them by using *characteristic views* [5, 14, 2, 17]. 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.

In this paper we will concentrate on the first step. That is, we will show how to the determine the correct view class given an image of an object. An example of how to solve the steps (2) and (3) can be found in [4].

2 Characteristic Views

Characteristic views can be found by analytically partitioning a viewing sphere centered at the object into aspects [20, 8, 19, 10]. 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" [16]. 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 to demonstrate our theory on view classification, since it is easy to implement. However, we realize of the shortcomings of this method and we plan to use an analytical approach in the future.

2.1 Sampling the Viewing Sphere

The viewing sphere is sampled using a hierarchical tessellation that subdivides the faces of an icosahedron recursively [16]. The centroid of each equilateral triangle is taken as a representative viewpoint for that triangle and the corresponding topological view, i.e. the aspect depends on this viewing position. For this paper, we used a resolution of 320 sample viewpoints, which is fine enough to partition the view sphere into fairly well defined clusters for our experiments.

2.2 Clustering Views

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

*This work was supported by a Pennsylvania State University Research Initiation Grant.

1063-6919/93 $03.00 © 1993 IEEE

which roughly the same segments were observed. This is a simple but effective criteria for classification.

Formally, let C be the set of sampled views $C = \{v_1, v_2, \ldots, v_n\}$. Each view is represented by a binary vector, each bit representing a model segment. A bit in the vector is set to 1 if the corresponding segment is observed in the view, and it is set to 0 otherwise. Then, we want find a partition $\{C_1, C_2, \ldots, C_c\}$ of the set C such that $\cup_{i=1}^{c} C_i = C$ and $C_i \cap C_j = \phi$, $\forall i \neq j$, and such that all the views in each subset C_i have the same segments.

This problem can be solved by using hierarchical clustering[7]. The algorithm starts with $c = 320$ singleton clusters. These clusters are then merged successively on the basis of a similarly function:

$$d(C_i, C_j) = \frac{x^T x'}{\| x \| \| x' \|} \tag{1}$$

where x, x' are the two binary vectors that need to be compared. This metric will cluster together those views that share in general, a high number of model segments.

3 Probabilistic Prediction Models

Most feature-based matching schemes assume that all the features that are potentially visible in a view cluster will appear with equal probability, and the resulting matching algorithms have to allow for "errors" without really understanding what they mean. PREMIO [3] is a CAD-based vision system that models some of the physical processes that can cause these errors. It uses CAD models of 3D objects and knowledge of surface reflectance properties, light sources, sensor characteristics, and feature detector algorithms to estimate the probability of the features being detectable as well as their attributes.

For each view cluster C of an object, the system PREMIO builds a ***probabilistic prediction model*** [4] by combining hundreds of views within the cluster. Next, we will briefly describe this model since it is the basis of our matching scheme.

A model M is 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 system ***predicts*** which features are detectable for a given configuration of light and sensor and a given image processing sequence. Then, given a set of n predictions within a view cluster, PREMIO approximates the detectability of a 2D feature by the frequency rate of its appearance. Two 2D features appearing in two different images are considered to be the same feature if they have a common 3D originating feature.

The set of labels L in the model M is formed by those 2D features that have high enough probability of being detected (above threshold t_f), as a whole or in pieces 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, PREMIO predicts which relationships among features would be detected and their attributes. The probability of a relation among a set of features to holding is approximated by the frequency rate of its appearance. The set of relational tuples R is formed for those relations among features in L such that they have high enough probability of holding (above threshold t_R). 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 $M = (L, R, f_U, g_S)$ 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 [4].

3.1 Features and Relations

The system PREMIO, in its current implementation, uses line segments as features. The edges extracted from an image are grouped perceptually to form interesting patterns [12]. These patterns constitute the relations among the features that are used in a consistent labeling scheme to match image features to model features. These arrangements are abstract, and their significance is determined by the three dimensional structure that they imply and by the amount of information they contribute to estimating the parameters in a transformation from three-dimensional model to the two-dimensional image. These features and relations are described in detail below.

Segments. Segments are characterized by their two vertices $s = (v_0, v_1)$. The segments in the model are uniquely labelled and during the matching process, it is desirable that there is a one-to-one mapping between the image segments and corresponding model segments. Each segment has 4 attributes associated with it, ie. the length of the segment, the midpoint of the segment (the x and y coordinate) and the orientation of the segment.

Junctions of 2 segments. Junctions of two segments, are junctions where two line segments meet, $J_2 = (s_0, s_1)$.

Junctions of 3 segments. Similarly, junctions of 3 segments are junctions where three line segments meet, $J_3 = (s_0, s_1, s_2)$.

Triples. A triple is an ordered set of three lines, $T = (l_0, l_1, l_2)$, with the lines traced clockwise, so that the triple has a well-defined inside. The convention for numbering the segments of a triple is fixed and is clockwise from inside. The angles are not measured, only their convexity is tested. Triples are good relations since they are plentiful in images of machined parts, but are not likely to be accidental.

3.2 Metric

Likewise a model, an image I is a quadruple $I = (U, S, f_U, g_S)$ where U is a set of image features or *units*, S is a set of relational tuples of units, f_U is the attribute-value mapping associated with U and g_S is the strength mapping associated with S.

In order to compare the attributes of the labels in a set L and those of the corresponding units in a set U we used the metric described below.

Let $h : L \rightarrow U$ be the feature correspondence mapping, that assigns labels in the model to units in the image. Then, the feature metric error for the mapping h, $E_{f_U}(h)$, is defined by:

$$E_{f_U}(h) = \rho(f_U \circ h, f_L|H) = \sum_{l \in H} \rho_{lu}(l, h(l)) \ , \qquad (2)$$

with

$$\rho_{lu}(l,u) = \sqrt{\rho_x^2(l,u) + \rho_y^2(l,u) + \rho_\lambda^2(l,u) + \rho_\alpha^2(l,u)}$$

where $u = h(l)$ and

$$\begin{array}{ll} \rho_x(l,u) = \frac{|\overline{x_m^l} - x_m^u|}{\sigma_{x_m}^l} & \rho_y(l,u) = \frac{|\overline{y_m^l} - y_m^u|}{\sigma_{y_m}^l} \\ \rho_\lambda(l,u) = \frac{|\overline{\lambda^l} - \lambda^u|}{\sigma_\lambda^l} & \rho_\alpha(l,u) = \frac{|\overline{\alpha^l} - \alpha^u|}{\sigma_\alpha^l} \end{array} ,$$

$((\overline{x_m}^l, \sigma_{x_m}^l), (\overline{y_m}^l, \sigma_{y_m}^l), (\overline{\lambda}^l, \sigma_\lambda^l), (\overline{\alpha}^l, \sigma_\alpha^l))$ are the attribute values of label l, $(x_m^u, y_m^u, \lambda^u, \alpha^u)$ are the attribute values of unit u, and H is the domain of the mapping h.

4 View Classification

Given an image, the objective is to identify an 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 will 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 Bayes' Theorem [18]:

$$P(C \mid I) = \frac{P(I \mid C)P(C)}{\sum_{i=1,\ldots,n} P(I \mid C_i)P(C_i)} \qquad (3)$$

4.1 Probabilistic Model

In order to apply equation (3), we need to compute the involved probabilities. In this section we describe a probabilistic model that can be used for this purpose.

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:

$$P(I|C) = \qquad (4)$$
$$(2\pi)^{-\frac{(d+4)}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu})\right]$$

where $\underline{x}$ is $(d+4) \times 1$ feature vector representing the image I. The feature vector 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, and the feature metric error 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.

The discriminant function for the i^{th} class is

$$g_i(x) = P(C_i|I)P(C_i) \ ,$$

or equivalently,

$$g_i'(x) = -1/2(\underline{x} - \underline{\mu}_i)^t \Sigma^{-1} (\underline{x} - \underline{\mu}_i) + \ln P(C_i) \ . \quad (5)$$

5 Experimental Protocol

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 the system using artificially generated data as well as real images. In this section we describe the experimental protocol used to test the system, based upon the one presented in [15].

5.1 Model Generation

The steps needed to generate the model are enumerated below.

1. Partition the viewing sphere C into a finite set of partitions $C_1, C_2, \ldots, C_n$. Each partition is called a cluster and $C_1 U C_2 U..C_n = C$
2. Each region C of the viewing space is a union of spherical sectors between two spheres. Each subregion is specified by a range of longitude $(\Phi_{C_{min}}, \Phi_{C_{max}})$ and latitude $(\theta_{C_{min}}, \theta_{C_{max}})$ angles and radius of the viewing spheres $(\rho_{C_{min}}, \rho_{C_{max}})$.
3. Corresponding to each cluster select a region $\mathcal{I}$ of the illumination space. The illumination space is defined in a manner identical to that of view space specification, i.e. by a patch which is bounded by $(\Phi_{I_{min}}, \Phi_{I_{max}})$ along the longitude and $(\theta_{I_{min}}, \theta_{I_{max}})$ angles along the latitude. The radius is specified $(\rho_{I_{min}}, \rho_{I_{max}})$.
4. The cluster is uniformly sampled. Let C_{i_s} denote a set of viewing positions and I_{i_s} be the corresponding set of light orientations. Here, the categorical variable v_i will take values which are limited by the cluster boundaries defined above.
5. Generate the predictions. For each pair $(C_i, I_i) \in C_s \times \mathcal{I}_s$, use the prediction module to predict the

subset of detectable labels L_{vi}, its associated relationship attribute mapping $f_{L_{vi}}$, the subset of dectectable relational tuples R_{vi}, and its associated relationship attribute mapping $g_{R_{vi}}$. The prediction module also generates the corresponding set of units U_{vi}, the associated mapping $f_{U_{vi}}$, the set of relational tuples S_{vi}, and the associated attribute mapping $g_{S_{vi}}$.

6. Obtain detectability frequencies. The previous step produced $N_v \times N_i$ different predictions. We approximate the probability of a label/relationship being detected, given that the view and the light are in the specified regions C_i and $\mathcal{I}_i$, 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 camera and light positions that were generated having uniform distributions as well as on the central limit theorem (provided that N_c and N_i are large enough).
7. Select desired detectability. Select the desired minimum label detectability t_f and the minimum relational detectability t_R and that the relational tuples have a detectability greater than t_R.
8. Combine the predictions. The $N_v \times N_i$ predictions are combined into a single model $M_i = (L, R, f_L, g_R)$ for each cluster such that the labels in L have a detectability greater than t_R.
9. Estimate the mean and covariance matrix of the probability distribution $P(I|C_i)$ by using the sample mean and covariance.

6 Experiments and Results

In our experiments we used a CCD camera with focal length 4.8 mm. and a resolution of 1.25901 mm./pixel × 1.18758 mm./pixel. The light is a point source of unpolarized light, of intensity 1 cd. The set of features L is made up of 2D-segments, projections of the 3D-segments forming the objects. The feature attribute mapping f_L assigns with each label l, four attributes: its midpoint image coordinates, x^l_m and, y^l_m, its length, λ^l, and its orientation α^l. Each label attribute value is given by a mean and a standard deviation representing the variations of the attribute among the different predictions used to obtain the model. The set of units U is the set of 2D-segments forming the image to be matched. The feature attribute mapping f_U associates to each unit u four attributes: its midpoint image coordinates, x^u_m and, y^u_m, its length, λ^u, and its orientation α^u.

Figure 1 shows images of *Cube3Cut* and *Fork*, two of the objects modeled in PREMIO. Figure 2 shows a few predictions for a cluster of Cube3Cut and a cluster of Fork.

We had used 5 clusters, of which 3 belonged to Cube3Cut (C_1, C_2, and C_3) and 2 belonged to Fork (F_1 and F_2). Figures 3 and 4 show visualizations of the sets of features of the models of Cube3Cut and Fork respectively. The features are drawn as segments with their mean attribute values. The numbers shown by the segments are the feature ID's and they indicate their relative detectability, with the lower the number, the higher the detectability.

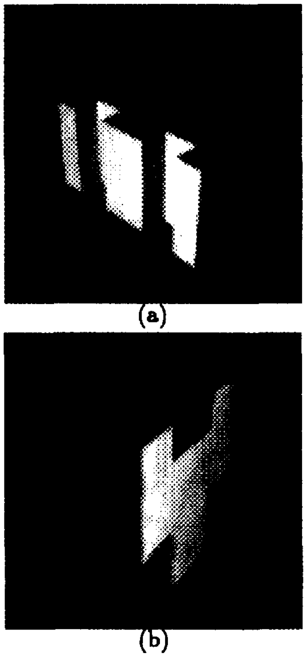


Figure 1: (a) Cube3Cut. (b) Fork.

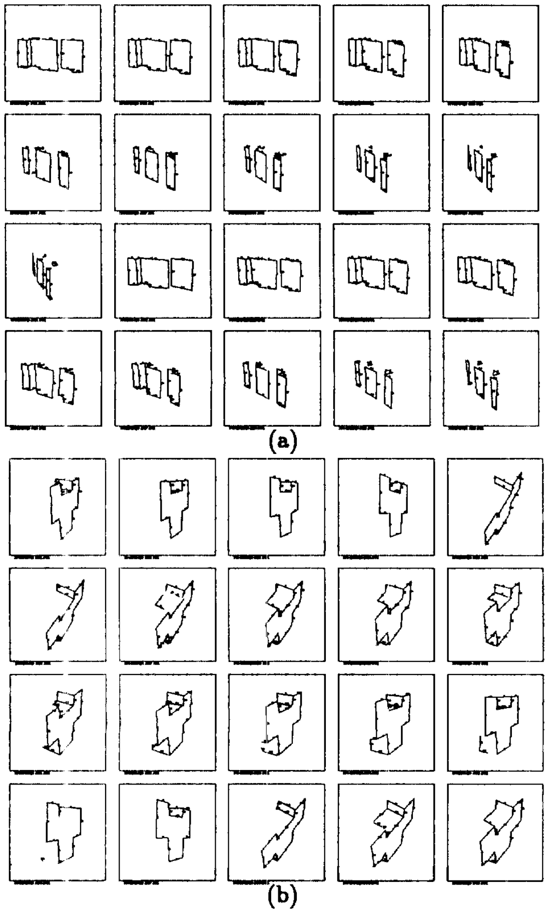
The classification results for 100 images of each cluster are summarized in Table 1 and 2. Table 1 gives the classifications obtained by selecting the cluster with the highest probability, while Table 2 gives the classifications obtained by selecting the cluster with the highest or second highest probability.

Table 1: Classification Results (First Choice).

	Classified as				
Belongs To	C_1	C_2	C_3	F_1	F_2
C_1	91	0	7	0	2
C_2	19	79	1	0	1
C_3	12	3	84	0	1
F_1	0	0	0	100	0
F_2	0	0	0	0	100

7 Conclusions

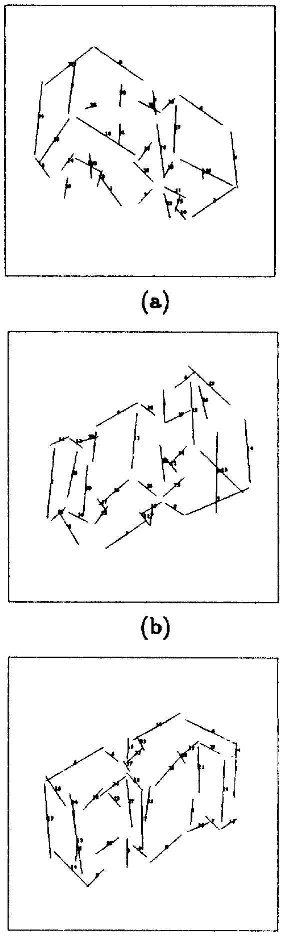
We presented a Bayesian approach to the view class determination problem. The view classes used contained probabilistic information that takes into account both geometrical and illumination characteristics. As shown in Tables 1 and 2 the test images matched best or second best to the correct view class in approximately 80% of the cases and above 90 %



(a)

(b)

Figure 2: (a) Cube3Cut: predicted images. (b) Fork: predicted images.



(a)

(b)

(c)

Figure 3: Cube3Cut models (segments). (a) C_1. (b) C_2. (c) C_3.

of the cases respectively. The images that failed to be correctly classified correspond to views near the boundaries of the clusters. Even though these views have the same segments as the rest of the views in their class, they look significantly different. This suggests that different definitions of clustering should be studied.

References

[1] R. Brooks. Symbolic reasoning among 3-D models and 2-D images. *Artificial Intelligence*, 17(1-3):285–348, 1981.

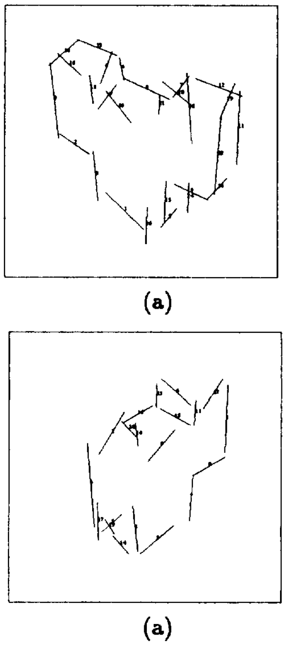
[2] O. I. Camps. *PREMIO: The Use of Prediction in a CAD-Model-Based Vision System.* PhD thesis, Department of Electrical Engineering, University of Washington, Seattle, Washington, 1992.

[3] O. I. Camps, L. G. Shapiro, and R. M. Haralick. PREMIO: An Overview. In *Proc. of the IEEE Workshop on Directions in Automated CAD-Based Vision*, pages 11–21, June 1991.

[4] O. I. Camps, L. G. Shapiro, and R. M. Haralick. Object Recognition Using Prediction and Probabilistic Matching. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1044–1052, Raleigh, North Carolina, July 1992.

[5] I. Chakravarty and H. Freeman. Characteristic views as a basis for three-dimensional object recognition. In *SPIE 336 (Robot Vision)*, pages 37–45, 1982.

[6] R. Chin and C. Dyer. Model-based recognition in robot vision. *Computing Surveys*, 18(1):67–108, March 1986.



(a)

(a)

Figure 4: Fork models (segments). (a) F_1. (b) F_2.

Table 2: Classification results (First/Second Choice).

	Classified as				
Belongs To	C_1	C_2	C_3	F_1	F_2
C_1	100	0	0	0	0
C_2	7	93	0	0	0
C_3	4	3	93	0	0
F_1	0	0	0	100	0
F_2	0	0	0	0	100

[7] R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* John Wiley, 1973.

[8] D. Eggert. *Aspect Graphs of Solids of Revolution.* PhD thesis, Department of Computer Science and Engineering, University of South Florida, Tampa, Florida, 1991.

[9] P. J. Flynn. *CAD-Based Computer Vision: Modeling and Recognition Strategies.* PhD thesis, Michigan State University, 1990.

[10] Z. Gigus and J. Malik. Computing the aspect graph of line drawings of polyhedral objects. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 654-661, April 1988.

[11] C. D. Hansen. *CAGD-Based Computer Vision: The Automatic Generation of Recognition Strateegies.* PhD thesis, The University of Utah, 1988.

[12] J. Henikoff and L. Shapiro. Interesting patterns for model-based matching. In *ICCV*, 1990.

[13] P. Horaud and R. Bolles. 3DPO: A system for matching 3-D objects in range data. In A. Pentland, editor, *From Pixels to Predicates*, pages 359–370. Ablex Publishing Corporation, Norwood, New Jersey, 1986.

[14] 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.

[15] T. Kanungo, M. Jaisimha, R. Haralick, and J. Palmer. An experimental methodology for performance characterization of a line detection algorithm. In *SPIE Conference on Optics, Illumination and Image Sensing for Machine Vision V*, pages 104–112, November 1990.

[16] M. Korn and C. Dyer. 3D-Multiview Object Representations for Model-Based Object Recognition. *Pattern Recognition*, 20(1):91–103, 1987.

[17] H. Lu, L. G. Shapiro, and O. I. Camps. A Relational Pyramid Approach to View Class Determination. In *Proc. IEEE Workshop on Interpretation of 3D Scenes*, pages 177–183, November 1989.

[18] A. Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, third edition, 1991.

[19] J. Ponce and D. Kriegman. Computing exact aspect graphs or curved objects: parametric surfaces. In *8th National Conference on AI*, pages 340-350, 1987.

[20] J. Stewman. *Viewer Centered Representations for Polyhedral Objects.* PhD thesis, Department of Computer Science and Engineering, University of South Florida, Tampa, Florida, 1991.