Probabilistic 3D Object Recognition

Ilan Shimshoni* and Jean Ponce
Department of Computer Science and Beckman Institute
University of Illinois, Urbana, IL 61801,
e-mail: ilans@ie.technion.ac.il.

Abstract: *A probabilistic 3D object recognition algorithm is presented. In order to guide the recognition process the probability that match hypotheses between image features and model features are correct is computed. A model is developed which uses the probabilistic peaking effect of measured angles and ratios of lengths by tracing iso-angle and iso-ratio curves on the viewing sphere. The model also accounts for various types of uncertainty in the input such as incomplete and inexact edge detection. For each match hypothesis the pose of the object and the pose uncertainty which is due to the uncertainty in vertex position are recovered. This is used to find sets of hypotheses which reinforce each other by matching features of the same object with compatible uncertainty regions. A probabilistic expression is used to rank these hypothesis sets. The hypothesis sets with the highest rank are output. The algorithm has been fully implemented, and tested on real images.*

1 Introduction

One of the major problems in computer vision is to recognize 3D objects appearing in a scene as instances from a database of models (see [6, 12] for reviews). Several recognition systems extract features such as points [21, 26, 27] or lines [4] from the image and match them to corresponding features in the database. They then verify each candidate hypothesis using other features in the image and finally rank the verified hypotheses.

*Ilan Shimshoni is now with the Department of Industrial Engineering and Management, Technion, Haifa 32000, Israel.

When constructing such a system the major problems to be addressed are: how to generate match hypotheses, in which order to process the hypotheses, how to verify them using additional image features, and how to rank the verified hypotheses. In our approach, we define a match hypothesis as the matching of a pair of edges or a trihedral corner (Figure 1) to corresponding feature sets in the model database. The hypotheses are ranked by computing the probability that each one is correct. In order to verify a hypothesis, we compute the pose of the object assuming the match is correct. Other hypotheses which match other image features to features in the same model should yield compatible poses if they correspond to the same object. The pose is transformed from a point in the pose space to a region of that space when uncertainty in the values measured in the image is taken into account, and we check whether hypotheses reinforce each other by testing whether their pose uncertainty regions intersect. We rank sets of hypotheses whose uncertainty regions intersect by the probability that the match of the whole set is correct, and output a small number of interpretations with the highest ranks. The main goal of our approach is to find a small number of probable interpretations which can then be verified by comparing the image to the hypothesized object using standard verification techniques.

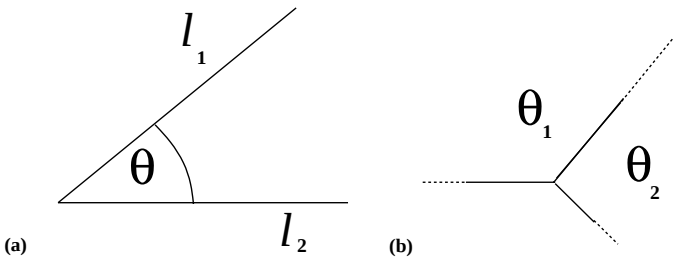


Figure 1: Image feature sets: (a) two lines in an image l_1 and l_2; the angle between the lines is θ; (b) a corner with three edges emanating from it.

2 Background and Approach

Since object recognition is one of the main problems in computer vision, a large body of work has addressed this problem (see [6, 12] for reviews). Here we focus on a few papers whose approach is related to ours.

Lowe [24] was the first to introduce the notion that a partial match of image to model features yields constraints on the position in the image of other features of the model due to rigidity constraints. This was generalized by Gaston and Lozano-Pérez [15] and Grimson and Lozano-Pérez [18] using the term *interpretation tree*. Classic examples of this approach are the algorithm of Ayache and Faugeras [4] for recognition of 2D objects, and the algorithm of Faugeras and Hebert [13] for recognition of 3D objects using range images.

In [14, 21], the *alignment* technique was presented. The algorithm chooses matches of the minimal size (three) which enables it to recover the pose of the object [1, 19]. Thus the position of other points in the image can be estimated and if found help verify the hypothesis. In order to produce reliable recognition systems of this kind the following problems have been addressed.

- What is the uncertainty in the pose given a possible match of three points in the model to points in the image whose position is uncertain [17].
- In which area around the projected model point should an image point be searched [2, 29].
- What size match is required to verify an hypothesis [16].
- In which order to test the hypotheses.

Probabilistic methods have been developed to rank hypotheses by their likelihood in

order to refrain from testing all possible match hypotheses. These methods are based on the probabilistic peaking effect [3, 7, 5, 9]. Informally this effect can be described as follows: when randomly choosing a direction from which to observe a pair of adjacent edges, and measuring the ratio of the lengths of the projections of the edges or the angle between them in the image, the values measured in the image have a high probability to be close to the real values measured in the model. In those papers, probability density functions for values measured in the image are estimated by uniformly sampling the viewing sphere. In addition, joint density functions for ratios of lengths and angles are estimated. The maximum of those density functions are always where the measured values equal the real values measured in the model. Using Bayes's rule, these probabilities can be used to estimate the probability that a set of features in the image match a set of features in the model. The probabilistic peaking effect is exploited in recognition systems presented in [26, 27, 5, 9, 31].

Other researchers take a different approach to recognition. They use statistical [20], and geometric [8, 11, 28] techniques to search the pose space looking for the pose and object which best describe the image without testing small matches of image and model features.

In this paper, we propose a 3D object recognition algorithm whose input is a set of straight lines extracted from a single image by an edge detector. In order to build a robust recognition system we have to account for the fact that the input is not perfect: objects may occlude each other, certain edges may not belong to any instance of any model in our model database, and not all edges appearing in the image will be extracted by the edge detector. We explicitly take all these factors into account.

We group the lines extracted from the image into two types of feature sets: pairs of adjacent lines and trihedral corners (Figure 1). For line pairs, we measure the lengths of edges and the angle between them, and measure only the angles between the lines for

trihedral corners. Therefore only pairs of edges whose visible part were fully recovered by the edge detector will yield valid hypothesis where as for trihedral corners that is not required. We have chosen these two types of features because under the scaled orthographic camera model that we use, they contain the minimal amount of information required to recover the viewing direction, and because the number of image and model features is linear in the number of vertices in the image and in our model database, reducing the number of hypotheses that have to be tested compared to testing all matches of vertex triplets in the image to vertex triplets in the model database (as in [21]).

In order to speed up the recognition process we rank the match hypotheses using the probabilistic peaking effect, and test them in the corresponding order. When the pose of the object is randomly selected, the probabilistic peaking effect says that values measured in the image such as ratios of lengths or angles between edges have a high probability to be close to the real values measured in the model. For the feature sets we have chosen, we measure for the pair of edges the ratio ρ of the lengths of the two edges and the angle θ between them, and measure two angles θ_1, θ_2 for the trihedral corner. We could have computed at recognition time for all the hypotheses the probability that the match is correct, but in order to speed up the algorithm we partition the (ρ, θ) and (θ_1, θ_2) spaces into rectangles and pre-compute the average probability of hypotheses with measured values within each rectangle. Under the scaled orthographic camera model, the only component of the pose affecting the ratios and angles in the image is the viewing direction which is parameterized as a point on the viewing sphere. The area on the viewing sphere where the measured values are within a rectangle correspond to the required probability. This area is bounded by curves on the viewing sphere where the ratio (iso-ratio curve) or the angle (iso-angle curve) is constant, and equal to the minimum or maximum values of each rectangle. We can accurately compute the desired probabilities by computing the area bounded by these curves. We use the probabilities computed for all model

feature sets to compute the probability that a match hypothesis is correct given that we measured certain values in the image. We compute these probabilities using Bayes's rule and account in the probabilistic model for self occlusion and imperfect edge data.

The model can be extended to incorporate more information about the image (e.g., the camera position and its relation to the surface on which the objects are placed), by reducing the space of legal poses to stable poses of the objects [22]. This would reduce the number of plausible hypotheses for each feature set considerably.

We verify a match hypothesis by computing the pose of the object, assuming the hypothesis is correct. We then try to find consistent match hypotheses with close poses that reinforce the hypothesis. We compute the viewing direction component of the pose of the object by tracing the iso-ratio or iso-angle curves on the viewing sphere which correspond to the ratios and angles measured in the image, and the viewing direction is the intersection point of those curves. The other components of the pose are easily obtained. This technique works for adjacent edges and trihedral corners, where previous work developed separate techniques for them [1, 21, 32].

Uncertainty in image data causes uncertainty in the pose. So in order to be able to decide whether two hypotheses reinforce each other, we estimate a region of the pose space which accounts for that uncertainty, so that only hypotheses whose regions intersect reinforce each other.

In the final stage of the algorithm we use a probabilistic expression to rank match hypotheses. Analysis of this expression shows that it has the following characteristics: hypotheses which would cause large errors in the measurements in the image are ranked lower than hypotheses with smaller errors, and combinations of feature sets which appear often in the model database such as rectangular faces are ranked lower than more rare feature combinations.

The rest of the paper is organized as follows. In Section 3 we develop a method

for computing probability density functions using equations we develop for angles and ratios. We use these probabilities for ranking match hypotheses in Section 4. We discuss pose estimation and estimate the effects of uncertainty on the pose in Section 5. We present our probabilistic expression to rank match hypotheses in Section 6. We show experimental recognition results in Section 7. Finally, a number of issues raised by our work and future research directions are discussed in Section 8.

3 Hypothesis Probability Computation

In this section we compute the probability that a given match hypothesis is correct. We compute probabilities and probability density functions for some observable image quantity (e.g., the ratio between lengths or an angle) to have a given value when the value (measured in the model) is given. We then extend this technique to joint probabilities and probability density functions for two image quantities, and use the results to rank match hypotheses.

3.1 Iso-ratio and Iso-angle Curve equations

When measuring the ratio of line lengths in an image or the angle between lines, we want to know from what viewing directions this image could be scanned, given a match between the lines and certain edges in the model. Consider two segments l_1 and l_2 in the image (Figure 1(a)) which are projections of edges $\mathbf{u}_1$ and $\mathbf{u}_2$ respectively in the model, such that the ratio between the length of l_1 and l_2 is ρ and the angle between them is θ.

The ratio between the lengths of the projections of $\mathbf{u}_1$ and $\mathbf{u}_2$ is ρ for viewing directions $\mathbf{v}$ which satisfy:

$$|\mathbf{u}_1 \times \mathbf{v}| - \rho|\mathbf{u}_2 \times \mathbf{v}| = 0. \tag{1}$$

Squaring this equation yields a quadratic equation in $\mathbf{v}$.

Viewing directions which satisfy:

$$(\mathbf{u}_1 \times \mathbf{v}) \cdot (\mathbf{u}_2 \times \mathbf{v}) - \cos\theta |\mathbf{u}_1 \times \mathbf{v}||\mathbf{u}_2 \times \mathbf{v}| = 0, \tag{2}$$

yield an angle θ between the projections of $\mathbf{u}_1$ and $\mathbf{u}_2$. Squaring this equation yields a equation of degree four in $\mathbf{v}$. Note that because only $\cos^2\theta$ appears in the equation, it also accounts for the curves corresponding to $\pi - \theta, 2\pi - \theta$ and $-(\pi - \theta)$. These must be identified and eliminated.

We trace these curves to gain a better understanding of the probabilistic peaking effect and use the area bounded by them to calculate probabilities. For tracing the curves we use an algorithm for tracing algebraic curves [23] which relies on homotopy continuation [25] to find all curve singularities and construct a discrete representation of the smooth branch curves. We add the constraint $|\mathbf{v}|^2 = 1$ to the equations derived earlier to insure that the viewing directions lie on the viewing sphere.

For reasons of symmetry, ratios are plotted using a $\log_2$ scale. In this way, the ratio and its inverse are symmetric around zero ($\log_2 \rho = -\log_2(1/\rho)$), and yield similar curves. We traced curves for values of ρ with equal intervals on the $\log_2 \rho$ scale. Figure 2(a) shows curves for values $\log_2 \rho$ between -2 and 2 where the the ratio in the model is 1. Interesting viewing directions are: viewing directions parallel to $\pm\mathbf{u}_1$ where $\mathbf{u}_1$ is foreshortened to a point and ρ is zero, viewing directions parallel to $\pm\mathbf{u}_2$ where $\mathbf{u}_2$ is foreshortened to a point and ρ is infinite, and viewing directions parallel to $\pm(\mathbf{u}_1 \times \mathbf{u}_2)$ where the viewing direction is orthogonal to the plane of $\mathbf{u}_1$ and $\mathbf{u}_2$ and the ratio is the "real" ratio. For ratios less than the "real" ratio there are two curves which surround the viewing direction parallel to $\pm\mathbf{u}_1$, for ratios greater than the "real" ratio there are two curves which surround the viewing direction parallel to $\pm\mathbf{u}_2$, and for the ratio equal to the "real" ratio ($\log_2(\rho) = 0$) the two curves intersect at the viewing directions parallel to $\pm(\mathbf{u}_1 \times \mathbf{u}_2)$.

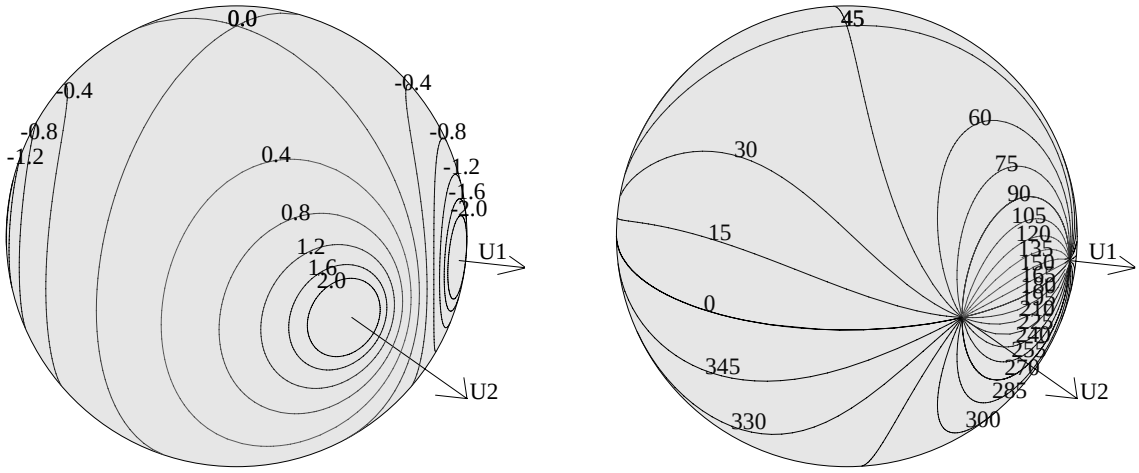
For the case when the "real" ratio is not one, consider a viewing direction $\mathbf{v}$ which satisfies

$$|\mathbf{u}_1 \times \mathbf{v}| - \rho|\mathbf{u}_2 \times \mathbf{v}| = 0 \tag{3}$$

for a pair of edges e_1 and e_2 such that $|\mathbf{u}_1| = |\mathbf{u}_2|$. Consider another pair of edges e_1' and e_2' such that $\mathbf{u}_1' = \lambda_1 \mathbf{u}_1$ and $\mathbf{u}_2' = \lambda_2 \mathbf{u}_2$. Substituting these values into (3) yields:

$$|\mathbf{u}_1' \times \mathbf{v}| - \rho\frac{\lambda_1}{\lambda_2}|\mathbf{u}_2' \times \mathbf{v}| = 0.$$

Thus the perceived ratio at the viewing direction $\mathbf{v}$ is $\rho\lambda_1/\lambda_2$ where λ_1/λ_2 is the "real" ratio between the lengths of e_1' and e_2'. This linear relationship enables us to concentrate on the case where the edges are of equal length and use this relationship to compute values for all other pairs of edges. No such relationship exists for angles.



(a) (b)

Figure 2: For two lines of equal length with 45° between them: (a) the viewing sphere with iso-ratio curves with $\log_2(\rho)$ in $[-2, 2]$; (b) the viewing sphere with iso-angle curves.

In Figure 2(b) we show curves for angles between 0 and 360° where the angle is defined as the angle between the projection $\mathbf{u}_1$ the projection of $\mathbf{u}_2$ in the counterclockwise

direction and the "real" angle between the two 3D edges is 45°. All the curves start at viewing directions parallel to $\pm\mathbf{u}_1$ and end at viewing directions parallel to $\pm\mathbf{u}_2$. This occurs because at those directions one of the edges has been foreshortened to zero length and therefore the angle between the two edges is not defined. However, by slightly changing the viewing direction any angle can be obtained. For angles less than the "real" angle, the curves go between $\mathbf{u}_1$ and $\mathbf{u}_2$ or between $-\mathbf{u}_1$ and $-\mathbf{u}_2$ starting for 0° from directions in the plane spanned by $\mathbf{u}_1$ and $\mathbf{u}_2$. At the "real" angle the two curves intersect like before at the viewing direction orthogonal to the $\mathbf{u}_1$,$\mathbf{u}_2$ plane. For angles greater than the "real" angle the curves go between $\mathbf{u}_1$ and $-\mathbf{u}_2$ or between $-\mathbf{u}_1$ and $\mathbf{u}_2$ ending for 180° with viewing directions which also lie in the $\mathbf{u}_1$,$\mathbf{u}_2$ plane. For angles θ greater than 180° the curve is the reflection of the curve for $360° - \theta$ through the origin. The probabilistic peaking effect is clearly demonstrated in this figure for both ratios and angles. The fraction of the sphere covered by ratios such that $-0.4 < \log_2 \rho < 0.4$ and by $30° < \theta < 60°$ is much larger than by other segments of the $\log_2 \rho$ and θ spaces of the same size.

3.2 Computing Probability Density Functions

Given the probability density function (p.d.f.) f, the value of the distribution function $F(a)$ is the probability that $x \leq a$. It is given by:

$$F(a) = \int_{-\infty}^{a} f(x)dx.$$

Conversely,

$$f(x) = F'(x),$$

therefore when one function is given, the other can be computed by integration or differentiation. In our case we first compute $F(a)$ for a ratio or an angle and then compute f

by numerical differentiation. The curve for a value of the ratio or an angle a is traced on the viewing sphere. This curve bounds the part of the viewing sphere for which $x \leq a$. The discrete points on the curve bound a polygon whose area is given by the following formula:

$$\text{area} = \sum_{i=1}^{n} \alpha_i - (n-2)\pi,$$

where n is the number of vertices of the polygon, and α_i is the spherical angle between two adjacent edges. A spherical angle is defined as the angle between the planes containing the great circles of the two edges. The area of the whole sphere is 4π, so to obtain probabilities the result must be divided by 4π. Examples of such curves for various types of distribution functions computed throughout this paper are shown in Figure 3.

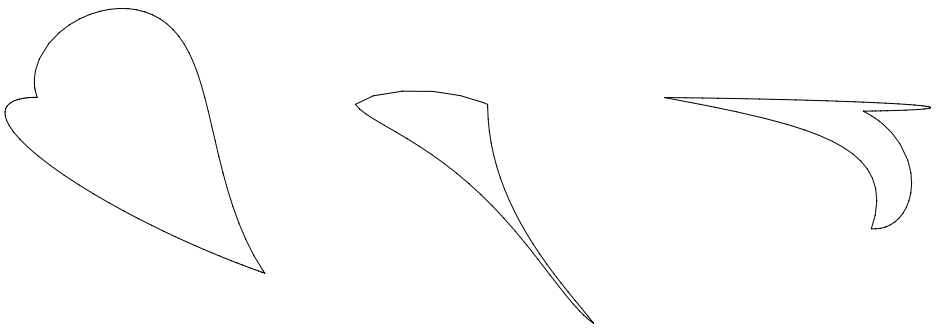


Figure 3: Regions on the viewing sphere whose areas are computed.

We can estimate $f(x)$ by:

$$f(x) = \lim_{\epsilon \to 0} \frac{F(x+\epsilon) - F(x-\epsilon)}{2\epsilon},$$

using the values computed for $F(x+\epsilon)$ and $F(x-\epsilon)$.

Figure 4(a) shows the probability density function for $\log_2 \rho$ for various values of the

original angle between the edges, where the original ratio is 1. A logarithmic scale is used to reflect the symmetry of the probability density function for ratios.

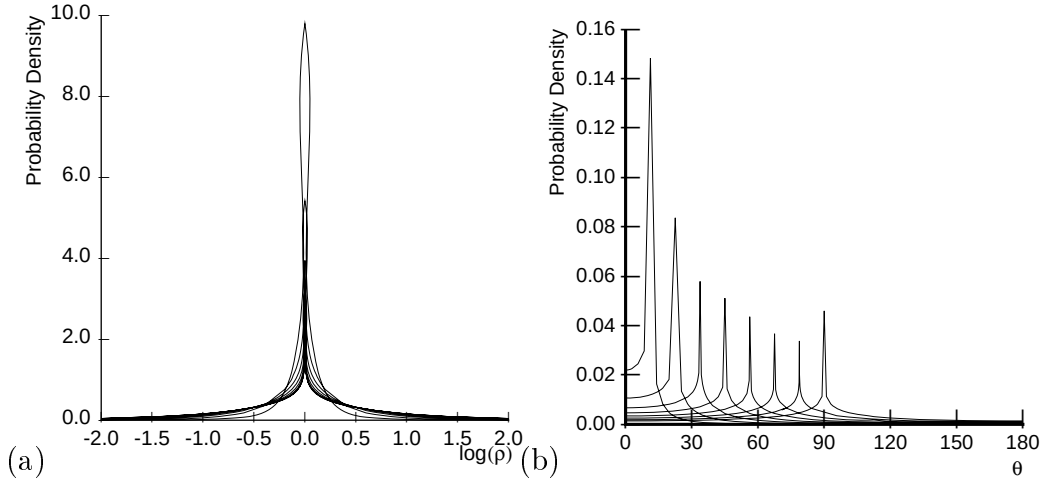


Figure 4: Probability densities for ratios and angles for various values of θ: (a) probability density for ratios; (b) probability density for angles.

Figure 4(b) shows the probability density function for the observed angle for various values of the original angle. In previous work Binford *et al.* [7], Ben-Arie [5] and Burns *et al.* [9] estimated probability density functions using ratios and angles computed of a uniform sample of the viewing sphere. However our method, which computes the areas on the viewing sphere produces the probability density functions exactly (up to the discretization error of the curves and the error due to finite differences which are negligible).

3.3 Computing Joint Probability Density Functions

In the previous section we showed how to compute probability density functions for one quantity measured in the image. However, when performing a partial match of image features to model features such as the ones in Figure 1, more than one value is measured.

Computing their joint probability density function can be used to rank the different match hypotheses for these image features. To demonstrate this we will examine the ratio-angle pair. The p.d.f. $f(\rho, \theta)$ and the distribution function $F(\rho, \theta)$ are related in the following ways:

$$F(\rho, \theta) = \int_0^{\rho} \int_0^{\theta} f(x, \alpha) dx d\alpha, \quad f(\rho, \theta) = \frac{\partial F}{\partial \rho \partial \theta}(\rho, \theta),$$

where $F(\rho, \theta)$ is the probability that the ratio between the lines in the image is less than ρ and the angle between them is less than θ. We estimate $f(\rho, \theta)$ as:

$$f(\rho, \theta) = \lim_{\substack{\mu \to 0 \\ \nu \to 0}} \frac{F(\rho + \mu, \theta + \nu) + F(\rho - \mu, \theta - \nu) - F(\rho + \mu, \theta - \nu) - F(\rho - \mu, \theta + \nu)}{2\mu 2\nu}, \tag{4}$$

where the numerator is the area bounded by the iso-ratio curves for $\rho \pm \mu$ and the iso-angle curves for $\theta \pm \nu$. The area is found by tracing the four curves, finding the intersection points between them using homotopy continuation, extracting the boundary of the region and computing its area.

3.4 Computing Joint Distribution Functions

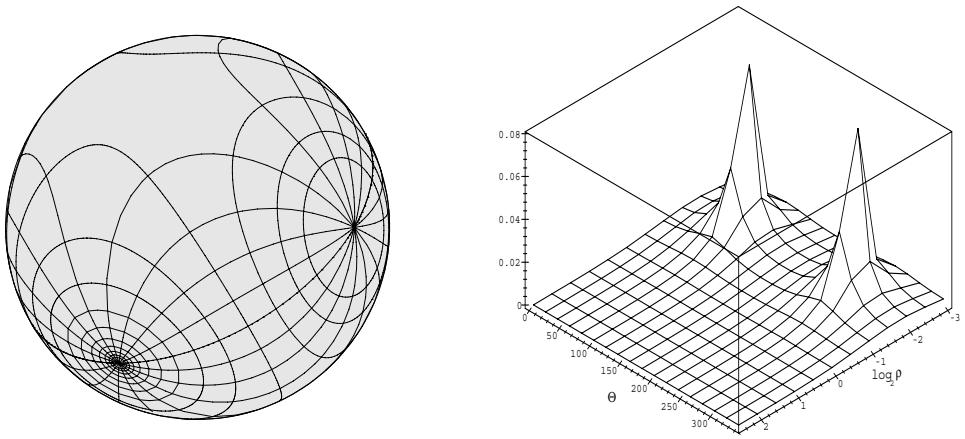
In order to rank match hypotheses we must be able to compute how likely the values measured in the image are when the match hypothesis is correct. The likelihood of a match hypothesis is measured by the value of the joint p.d.f. which was computed in the previous section for the values measured in the image. This can be done during the recognition process for all the match hypotheses. However in order to speed up the recognition process we build look-up tables off-line. The $(\log_2 \rho, \theta)$ and (θ_1, θ_2) spaces are divided into rectangles (12×18, and 18×18 rectangles respectively) and for each rectangle the average value of the joint p.d.f. is computed. At recognition time the value of the joint p.d.f. is found in the appropriate entry of the look-up table.

To perform that we have to compute for two adjacent edges e_1 and e_2 the probability

$$P(\rho_1 < \rho < \rho_2, \theta_1 < \theta < \theta_2), \tag{5}$$

that when viewing these edges from a randomly selected viewing direction the ratio of the lengths ρ will be between ρ_1 and ρ_2 and the angle between the edges will be between θ_1 and θ_2. We will denote this region of the (ρ, θ) space by $R(\rho_1, \rho_2, \theta_1, \theta_2)$. Using the technique described in the previous section we trace the curves of ρ_1, ρ_2, θ_1 and θ_2 and find the area bounded by them. Because for every value of ρ or θ, there are two curves, two such areas exist. We have divided the $(\log_2 \rho, \theta)$ space into identical rectangles. The $\log_2 \rho$ dimension was truncated at $-k$ and k where k is an arbitrary limit set taking into account the maximum and minimum lengths of edges assumed to be extracted by the edge detector, and using those values to estimate the maximum and minimum ratios of lengths. Dividing $P(\rho_1 < \rho < \rho_2, \theta_1 < \theta < \theta_2)$ by the area of the rectangle in the $(\log_2 \rho, \theta)$ space yields the average joint p.d.f. value for ratios and angles within that rectangle. We have traced the corresponding regions on the viewing sphere of each rectangle, producing a tessellation of the viewing sphere shown in Fig 5(a). The areas of the regions which represent $P((\rho, \theta) \in R(\rho_1, \rho_2, \theta_1, \theta_2))$ are plotted in Figure 5(b).

For trihedral corner feature sets curves for the two angles are plotted producing a tessellation of the viewing sphere shown in Figure 6(a). The areas of the regions which represent $P((\theta_1, \theta_2) \in R(\theta_{11}, \theta_{12}, \theta_{21}, \theta_{22}))$ were plotted in Figure 6(b). Corners are characterized by the three angles between the three pairs of edges. Although the two sets of curves (such as the set in Figure 2(b)) are determined by the first two angles, the third angle influences where the two sets of curves will intersect and thus the area of the regions (if they exist). To demonstrate how different the results can be, we have computed the joint probability function of a number of corners, and display in Figure 7(top) the graph of the probability function. The bottom of the figure shows the regions which have zero



(a) (b)

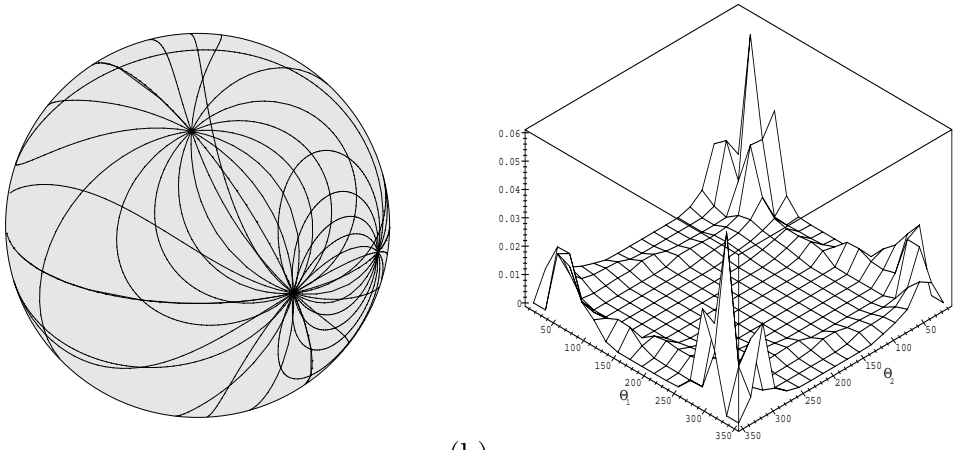
Figure 5: The joint probability function of ratios and angles for a pair of edges where the original values are $(1/3, 90°)$: (a) tessellation of the viewing sphere into ratio/angle regions; (b) a graph of the joint probability function for ratios and angles as estimated by the area of the regions in (a).

and non-zero probability. This shows that for a given corner there doesn't always exist a viewing direction for every pair of angles, where as in the ratio/angle case there exist viewing directions for all values of ρ and θ. This enables us to discard some of the hypotheses for trihedral corners because there is no viewing direction which would yield those angles for some of the model feature sets.

In the ratio/ratio case the feature set includes three edges. We could then measure in the image in addition to the two ratios, two angles. This means that the ratio/ratio case can be modeled as an angle/angle and two ratio/angle feature sets. So when a ratio/ratio feature set occurs in the image it is modeled as a combination of its simpler components.

3.5 Dealing with Occlusion

Up until now our analysis did not account for the fact that the features we are studying belong to solid objects and these objects could partially or totally occlude these features.



(a) (b)

Figure 6: (a) Tessellation of the viewing sphere into angle/angle regions; (b) a graph of the joint probability function for pairs of angles as estimated by the area of the regions in (a).

The features can be occluded by features of the same object or by other objects. Modeling the effects of occlusion between different objects is very difficult without having prior knowledge about the location of the objects and the camera in the scene.

Self occlusion on the other hand can be modeled exactly by dealing with the following two problems: for both types of feature sets the vertex at which the edges meet (the corner) must be visible, and when computing the ratio of lengths in two edge feature sets, only the length of the visible part of the projection of the edge should be used. We determine the visibility of the features using the aspect graph of the object.

As the visibility of features only changes on the viewing sphere on critical curves, we can analyze the visibility of the features in the feature set by studying a representative aspect from each non-critical region, which are bounded by the critical curves, and determine where the features are visible and which critical curves bound those regions. Regions in which the corner is not visible are removed from the probability distribution

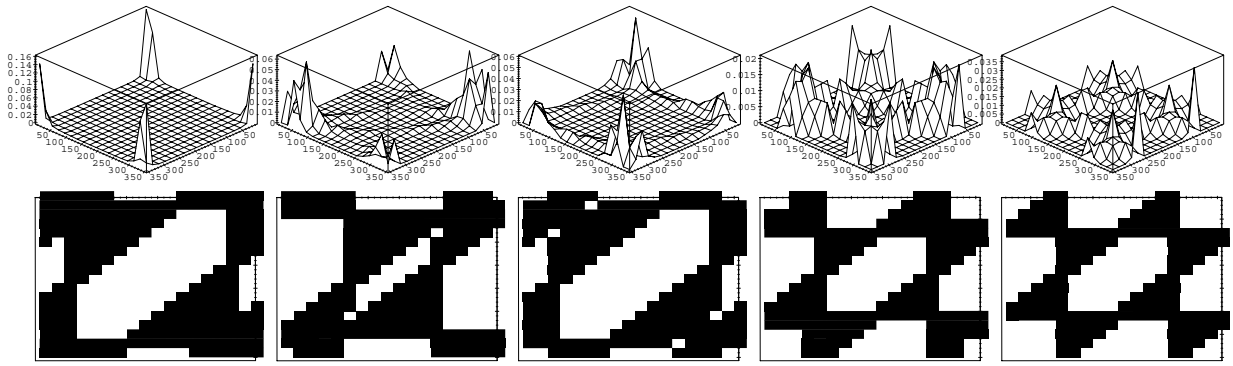


Figure 7: For trihedral corners with the following angles: $(10°, 20°, 25°)$, $(40°, 60°, 40°)$, $(40°, 60°, 80°)$, $(80°, 80°, 80°)$ and $(90°, 90°, 90°)$ graphs of joint probability functions and the regions which have zero and non-zero probability.

computed earlier, and for areas in which one or both the edges are partially occluded we replace the curves traced for the ratio of lengths by new curves using the following derivation.

Given two edges e_1 and e_2 emanating from a visible vertex p which are shown in Figure 8, we characterize the viewing directions $\mathbf{v}$ such that the ratio of the lengths of the visible part of their projections is ρ. So if the projections of e_1 and e_2 intersect the projections of e_1' and e_2' at $p + t_1\mathbf{u}_1$ and $p + t_2\mathbf{u}_2$ respectively, (1) is modified into:

$$t_1|\mathbf{u}_1 \times \mathbf{v}| - \rho t_2|\mathbf{u}_2 \times \mathbf{v}| = 0. \tag{6}$$

To characterize the intersections of straight lines in a simple way, we use Plücker coordinates, which describe a line by two orthogonal vectors $(\mathbf{a}, \mathbf{b})$ where $\mathbf{a}$ is some direction along the line, and if p is a point on the line, $\mathbf{b} = p \times \mathbf{a}$. We use the following property: two straight lines with Plücker coordinates $(\mathbf{a}_1, \mathbf{b}_1)$ and $(\mathbf{a}_2, \mathbf{b}_2)$ intersect if and only if $\mathbf{a}_1 \cdot \mathbf{b}_2 + \mathbf{a}_2 \cdot \mathbf{b}_1 = 0$. We write that a line D_1'' passing through $p + t\mathbf{u}_1$ with direction $\mathbf{v}$ intersects the supporting line $D_1' = (a_1, b_1)$ of e_1'. The line D_1'' has Plücker

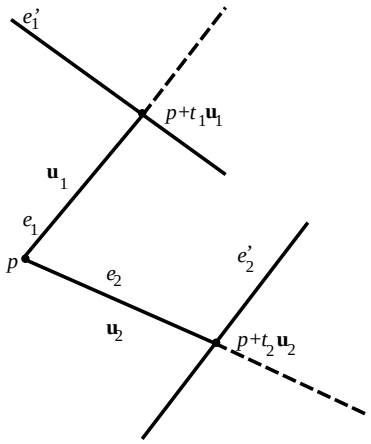


Figure 8: Two adjacent edges e_1 and e_2, starting from a visible vertex p in directions $\mathbf{u}_1$ and $\mathbf{u}_2$ respectively, intersect edges e_1' and e_2' at $p + t_1\mathbf{u}_1$ and $p + t_2\mathbf{u}_2$ respectively.

coordinates $(\mathbf{v}, (p + t_1\mathbf{u}_1) \times \mathbf{v})$. Writing that $D_1^{'}$ intersects $D_1^{''}$ we obtain:

$$\mathbf{a}_1 \cdot (t_1\mathbf{u}_1 \times \mathbf{v} + p \times \mathbf{v}) + \mathbf{b}_1 \cdot \mathbf{v} = 0.$$

Similarly we obtain for the intersection of e_2 and e_2':

$$\mathbf{a}_2 \cdot (t_2\mathbf{u}_2 \times \mathbf{v} + p \times \mathbf{v}) + \mathbf{b}_2 \cdot \mathbf{v} = 0.$$

Substituting the equations for t_1 and t_2 obtained from these equations into (6) yields:

$$\frac{-\mathbf{b}_1 \cdot \mathbf{v} - \mathbf{a}_1 \cdot (p \times \mathbf{v})}{\mathbf{a}_1 \cdot (\mathbf{u}_1 \times \mathbf{v})} |\mathbf{u}_1 \times \mathbf{v}| - \rho \frac{-\mathbf{b}_2 \cdot \mathbf{v} - \mathbf{a}_2 \cdot (p \times \mathbf{v})}{\mathbf{a}_2 \cdot (\mathbf{u}_2 \times \mathbf{v})} |\mathbf{u}_2 \times \mathbf{v}| = 0. \tag{7}$$

Squaring this equation yields a degree six homogeneous equation in $\mathbf{v}$. If only one of the edges is partially occluded the equation simplifies to an equation of degree four.

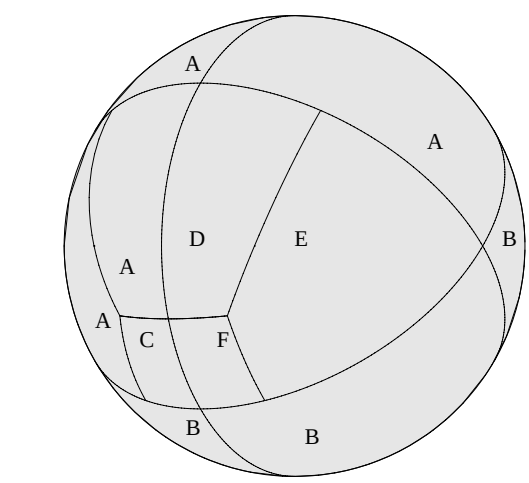
To demonstrate the effect of occlusion we have chosen two edges of an L shaped object whose aspect graph is shown in Figure 9(a). We labeled the non-critical regions of the aspect graph with letters $A - F$ according to the state of occlusion of the two edges and show representative views of the object in Figure 9(b). In regions labeled by A both

edges are totally visible therefore (1) is used to trace the iso-ratio curves. In regions labeled by B the vertex is occluded, and therefore these regions are discarded. In C one of the edges is partially occluded and in D and E the other edge is partially occluded, each time by a different edge. For these regions (7) is used to trace the iso-ratio curves. In region F the corner is occluded therefore this region is also discarded.

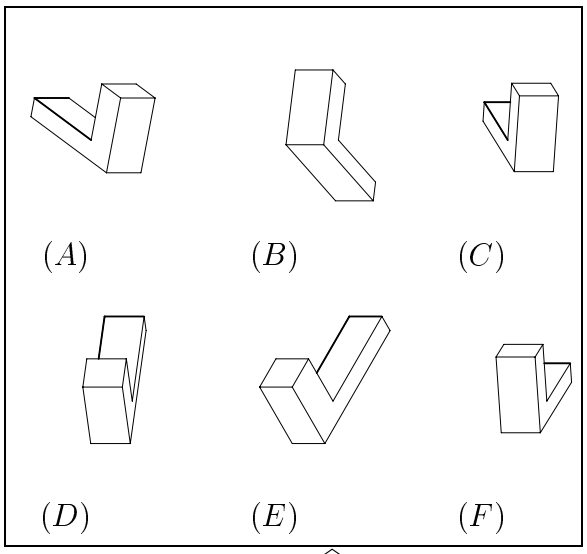
In Figure 9(c) the aspect graph and the tessellation of the sphere into regions are traced on the viewing sphere. It is interesting to note the effect of crossing a critical curve on the iso-ratio curve. On the boundary between regions A and E the iso-ratio curve is continuous where as on the boundary between regions A and D or A and C it is not. In the former when the viewing direction is part of the critical curve the projection of the occluding edge touches a vertex of one of the edges in the feature set and therefore does not occlude it at all. In the latter when the critical curve is passed the edge changes from being totally visible to being partially occluded changing the ratio of the visible parts of the projections of the edges instantaneously. In region F the corner is occluded and on its boundaries e_1 (C) and e_2 (D and E) become totally occluded yielding sets of dense iso-ratio curves with values converging on zero and infinity respectively. The probability distribution function is shown in Figure 9(d) (compare to Figure 5 which disregards occlusion). Similar probability distributions have been computed for the case of the trihedral corner. In that case regions on the viewing sphere where the corner is not visible were discarded. In the other regions the regular iso-angle equations were used.

4 Ranking Match Hypotheses

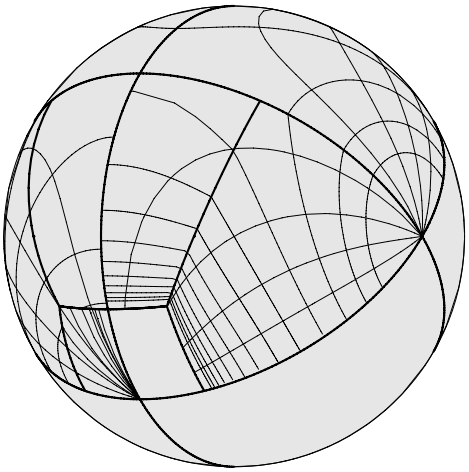
In this section we use the techniques presented in the previous section for ranking match hypotheses. We build look-up tables of lists of hypotheses sorted by probability off-line and during the recognition stage the sorted hypotheses are tested. We will assume for



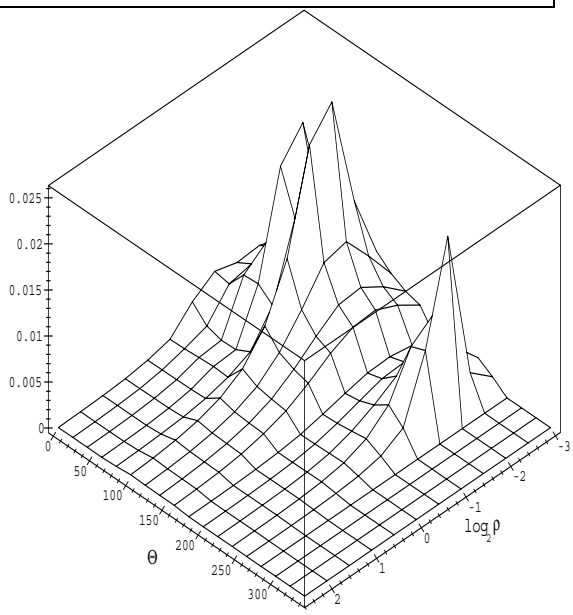
(a)



(b)



(c)



(d)

Figure 9: (a) the aspect graph of the L shaped object with the regions marked by letters A-F; (b) views of the object: (A) both edges are visible; (B) at least one of the edges is not visible; (C) one of the edges is partially occluded; (D)-(E) the other edge is partially occluded by two different edges; (F) the corner is occluded; (c) tessellation of the viewing sphere into regions; (d) a graph of the joint probability function as estimated by the area of the regions in (c).

now that all feature sets detected in image are due to objects in the model database. We shall relax that assumption in the next section.

4.1 Building Look-up Tables

In the preprocessing stage, we construct two look-up tables $T_1(\rho, \theta)$ and $T_2(\theta_1, \theta_2)$ for the ratio/angle and angle/angle pairs respectively. The tables are built by computing joint probability distributions which were described in Section 3.4 for all feature sets of all models in the model database.

We use these probabilities to determine the induced probabilities on the identity of the object. We denote by E a feature set measured in the image with values within a certain region of the (ρ, θ) or (θ_1, θ_2) space, by M_i the event in which the i^{th} model from the database appears in the image, and by $H_j^{(i)}$ the hypothesis that the j^{th} model feature set of the i^{th} model matches E. We would like to compute $P(H_j^{(i)}, M_i|E)$ which is the probability that hypothesis $H_j^{(i)}$ is correct if E was measured in the image. Using Bayes' law

$$P(H_j^{(i)}, M_i|E) = \frac{P(E, H_j^{(i)}|M_i)P(M_i)}{P(E)}.$$

$P(E, H_j^{(i)}, |M_i)$ is the probability that features associated with hypothesis $H_j^{(i)}$ were visible and that the values measured in the image were within the region of E assuming that object M_i is visible in the scene. This probability was computed in the previous section. $P(E)$ is the sum of all the probabilities of all the hypotheses in which E can be measured. Therefore,

$$P(E) = \sum_m \sum_j P(E, H_j^{(m)}|M_m)P(M_m).$$

Combining these two equations yields:

$$P(H_j^{(i)}, M_i|E) = \frac{P(E, H_j^{(i)}|M_i)P(M_i)}{\sum_m \sum_j P(E, H_j^{(m)}|M_m)P(M_m)}. \quad (8)$$

$P(M_i)$ can be determined using any prior knowledge we have about the likelihood of a model to appear in the image. We keep in tables an entry for each E. In each entry there is a list of hypotheses due to E. $P(H_j^{(i)}, M_i|E)$ is computed for the hypotheses in the list. This is repeated for all the entries in the tables and the lists are sorted by probability.

It is important to note that the regions in the (ρ, θ) or (θ_1, θ_2) spaces do not have to be of equal size. Moreover, the recognition algorithm will perform better if regions with multiple probabilistic peaks are divided into smaller regions with one peak in each. Thus each hypothesis would have a higher probability in the subregion in which its peak is located and a lower probability in the other subregion, where as if only one region existed all the hypotheses would have the same average probability. In general, the (ρ, θ) or (θ_1, θ_2) space can be partitioned using a quad-tree approach where a region is divided into four regions only if the order of the hypotheses of the subregions is different than the order of the hypotheses of the parent region.

4.2 Dealing With Uncertainty

The input to the recognition phase of the algorithm is the result of edge and corner detection performed on the image. There is uncertainty in this data which has to be modeled in order to design a robust algorithm. We identify three types of uncertainty which have to be dealt with: certain feature sets which appear in the image will not be recovered by the edge and corner detectors, the image features might not match any model features, and the values measured for image features are themselves uncertain.

We denote by $P_d(E,H)$ the probability that a feature set which appears in the image, for which we measure values E, and for which the hypothesis H is correct will be detected by the edge and corner detectors. For each hypothesis evidence pair $P_d(E,H)$ can be in principle obtained empirically by testing the performance of edge and corner detectors on many typical scenes. In general, feature sets appearing on the silhouette of the object will have a higher $P_d(E,H)$ than internal features.

Incorporating $P_d(E,H)$ into (8) yields:

$$P(H_j^{(i)}, M_i|E) = \frac{P(E, H_j^{(i)}|M_i)P(M_i)P_d(E, H_j^{(i)})}{\sum_m \sum_l P(E, H_l^{(m)}|M_m)P(M_m)P_d(E, H_l^{(m)})}.$$

The second type of uncertainty deals with feature sets which are not "legal" feature sets. Image features may not match model features for various reasons: the features were not correctly recovered (e.g. only part of an edge was recovered), the features belong to several occluding objects, or the features belong to objects not in the database or to the background. The average number of "illegal" feature sets P_{il} in a scene depends on the types of scenes which the recognition process analyzes and on the quality of the edge and corner detectors. However, P_{il} can be estimated empirically by testing the system on images of typical scenes. Different types of environments will produce different values for P_{il}. In order to incorporate this information into $T_1(\rho,\theta)$ and $T_2(\theta_1,\theta_2)$ tables we compute the probability that we measured values within a region in the value space due to random features. For example if we assume that the length of the lines and the angles between them are uniformly distributed, the distribution of $\log_2 \rho$ which is the logarithm of the ratio of the lengths of two lines with uniformly distributed lengths, is $\frac{1}{2}e^{-|\log_2 \rho \ln 2|}$. The probability for a pair of edges to be in a rectangle in the (ρ,θ) space is:

$$P((\rho,\theta) \in R(\rho_1,\rho_2,\theta_1,\theta_2)) = \frac{1}{2}|e^{-|\log_2 \rho_1 \ln 2|} - e^{-|\log_2 \rho_2 \ln 2|}|\frac{(\theta_2-\theta_1)}{2\pi}.$$

The probability for a corner to be in a rectangle in the (θ_1, θ_2) space is:

$$P((\theta_1, \theta_2) \in R(\theta_{11}, \theta_{12}, \theta_{21}, \theta_{22})) = \frac{(\theta_{12} - \theta_{11})}{2\pi} \frac{(\theta_{22} - \theta_{21})}{2\pi}.$$

In each entry in the tables a new hypothesis is added to the list, which represents the "illegal" feature set and whose probability is the probability calculated above multiplied by P_{il}.

The last type of uncertainty is in the values measured in the image. For the two-edge case we assume that there is uncertainty in vertex position of the three vertices. We assume the six coordinates $\mathbf{a} = (x_1, y_1, x_2, y_2, x_3, y_3)$ have a normal distribution with mean $\mu_i = a_i$ and variance σ^2 which can be determined empirically. This induces a probability distribution on ρ and θ which enables us to compute the probability that the actual measured values are in a region of the (ρ, θ) space. Thus

$$P(H|E) = \sum_k P(H|E_k)P(E_k|E) = \sum_k P(H|E_k) \frac{P(E|E_k)P(E_k)}{\sum_j P(E|E_j)P(E_j)},$$

where $P(E|E_k)$ is the probability that E is in the k^{th} region of the (ρ, θ) space. $P(E|E_k)$ decreases rapidly as the distance from E to the region E_k increases, and the rate of the decrease accelerates as σ^2 gets smaller. Therefore for most regions E_k except the ones close to E, $P(E|E_k)$ is negligible and $P(H|E)$ is computed by summing over the contributions of a small number of regions.

5 Pose and Pose Uncertainty Estimation

For all correct hypotheses which match features from an instance of an object to features in the model, the pose recovered for all of these match hypotheses should be the same. Therefore we use pose estimation to find sets of hypotheses which produce the same pose to reinforce the recognition hypothesis. Uncertainty in the values measured in the

image induces uncertainty in the pose and has to be accounted for when testing the compatibility of match hypotheses.

5.1 Pose Estimation

The problem of estimating the pose from three points [1, 19, 21] or a trihedral corner [32] has been extensively studied. We present here a simple approach which deals with both types of feature sets. We regard the pose of an object in weak perspective projection as a combination of the following four components:

- a viewing direction $\mathbf{v}$ which is a point on the viewing sphere,
- a rotation of the image by δ degrees about the viewing direction,
- a scale s,
- a translation $\mathbf{t}$ in the image.

Therefore the pose is a point in the six dimensional space $\{S^2 \times [0 \cdots 2\pi] \times [s_{min} \cdots s_{max}] \times \mathbb{R}^2\}$ where S^2 is the unit viewing sphere, s_{min} and s_{max} are the minimum and maximum assumed scales respectively, and the translation $\mathbf{t}$ must be a vector in the image. The projection p_i of a point p of the object in the image is:

$$p_i = sR(\delta)(p \cdot \mathbf{v}_2, p \cdot \mathbf{v}_3) + \mathbf{t}, \tag{9}$$

where $\mathbf{v}, \mathbf{v}_2, \mathbf{v}_3$ is an orthonormal basis of $\mathbb{R}^3$ and $R(\delta)$ is a rotation by δ degrees.

We will now show what components of the pose can be recovered from angles and lengths measured in the image given an hypothesis which matches a minimal number of features in the image to features in the model. Each measured angle or ratio imposes a one-dimensional constraint on possible viewing directions. In order to determine $\mathbf{v}$ two such constraints are needed. Two pairs of types of curves were considered: the ratio/angle

pair and the angle/angle pair. The ratio/ratio pair is not considered because in order to measure two ratios we would need an hypothesis which contains additional matched features which we are trying to avoid. $\mathbf{v}$ is obtained as the intersection points between two curves. The degrees of the ratio and angle curves are 2 and 4 respectively. Therefore the number of intersection points found for the ratio/angle and angle/angle pairs will be at most 8 and 16 respectively. However as shown in Section 3 the angle equations generate curves for $\theta, \pi - \theta, 2\pi - \theta$ and $-(\pi - \theta)$. As most of the solutions are for the other angles the number of real solutions is much less. Some of the remaining solutions can be eliminated by visibility considerations (i.e., the features are occluded from that viewing direction). The other components of the pose are determined using standard 2D pose estimation. The rotation angle δ is determined for both pairs by rotating the results of applying the projection of the object in direction $\mathbf{v}$ until the corresponding edges are parallel to each other. The scale and translation can not be obtained for the angle/angle case. Therefore only in the case of the ratio/angle pair they are recovered.

5.2 Efficient Pose Uncertainty Estimation

We compute the pose p of the object assuming a match hypothesis h is computed using the technique described in Section 5.1. p is a function of the vertex positions $\mathbf{a}$ measured in the image. Using Taylor expansion, the effect of a small uncertainty $\boldsymbol{\xi}$ in $\mathbf{a}$ on the pose can be estimated by:

$$p(\mathbf{a} + \boldsymbol{\xi}) \approx p(\mathbf{a}) + \nabla p(\mathbf{a})\boldsymbol{\xi}.$$

As the uncertainty is small, the contributions of higher derivatives of the pose function can be neglected. For the ratio/angle case all six components of the pose are recovered using three coordinate pairs. Thus $\nabla p(\mathbf{a})$ is a 6×6 matrix. Assuming that the uncertainty for each coordinate pair is bounded by ϵ, the maximum uncertainty for a component p_ν

of pose will be when $\boldsymbol{\xi}$ has the following values:

$$\begin{cases} \boldsymbol{\xi}_{2i-1} = \epsilon \frac{\partial p_\nu}{\partial a_{2i-1}} \left(\frac{\partial p_\nu}{\partial a_{2i-1}}^2 + \frac{\partial p_\nu}{\partial a_{2i}}^2 \right)^{-1/2}, \\ \boldsymbol{\xi}_{2i} \quad = \epsilon \frac{\partial p_\nu}{\partial a_{2i}} \left(\frac{\partial p_\nu}{\partial a_{2i-1}}^2 + \frac{\partial p_\nu}{\partial a_{2i}}^2 \right)^{-1/2}, \end{cases}$$

where i denotes the i^{th} coordinate pair. Thus for each coordinate pair the vector $(\boldsymbol{\xi}_{2i-1}, \boldsymbol{\xi}_{2i})$ points in the direction of $\nabla p_\nu(\mathbf{a}_{2i-1}, \mathbf{a}_{2i})$. The derivatives are computed numerically. The perturbed pose is computed using the multivariate Newton-Raphson algorithm with the unperturbed pose given as the initial guess. For most components of the pose computing the uncertainty is simple. However for the viewing direction component $\mathbf{v}$ of the pose we parameterize the pose as $\mathbf{v} = \alpha\mathbf{v}_1 + \beta\mathbf{v}_2 + \gamma\mathbf{v}_3$ where $\mathbf{v}_1$ is the viewing direction for the unperturbed input, $\alpha = \sqrt{(1 - \beta^2 - \gamma^2)}$, and the uncertainty is measured in radians in the $\mathbf{v}_2$ and $\mathbf{v}_3$ directions. So if the viewing direction uncertainties are $\arcsin\beta$ and $\arcsin\gamma$, the viewing direction component in the pose uncertainty region is:

$$\{\mathbf{v} \in S^2 : \quad |\mathbf{v} \cdot \mathbf{v}_2| < \beta, \quad |\mathbf{v} \cdot \mathbf{v}_3| < \gamma\}.$$

In order to check if two pose uncertainty regions have a non-empty intersection, the regions of all components of the pose (viewing sphere,rotation,scale,translation) have to intersect. This works for pairs of ratio/angle hypotheses, but for angle/angle hypotheses only the viewing direction and rotation components can be recovered directly and more information is needed to recover the scale and translation components. By adding to the feature set the position and uncertainty of the corner of the other feature set, the scale and translation components of the uncertainty region are also recovered Then the intersection of all the components is tested.

To demonstrate the variability of the pose uncertainty for different model feature sets, we tabulate the pose uncertainty for them in Table 1 where the ratio and the angle in the image were 1.4 and 315° respectively. The sizes of nearly all the components of the pose

region do not change much between the different examples. Only the size of the viewing sphere component changes dramatically for the different examples. The closer the ratio and angle measured in the model are to the values measured in the image the larger the viewing sphere component is. This phenomenon, is another aspect of the probabilistic peaking effect. Thus, hypotheses with low probability (viewed values very different from real values) will have smaller pose uncertainty regions which will reduce the chance that there will be an intersection between the pose uncertainty regions of two such hypotheses, reducing the chance for false a positive recognition result.

Ratio	Angle	Viewing Sphere	Rotation	Scale	Translation
1.0	280.0°	0.011	0.092	0.139	0.0080
0.3	90.0°	0.001	0.101	0.147	0.0329
1.4	315.0°	0.085	0.184	0.158	0.0459
0.3	315.0°	0.001	0.109	0.173	0.0099
1.4	90.0°	0.009	0.124	0.114	0.0142

Table 1: Tabulation of the size of components of the pose uncertainty region for a given image feature set whose ratio and angle are 1.4 and 315.0° respectively for several model feature sets with different ratios and angles. Note that for measured values equal to the real values the uncertainty region is much larger (up to a factor of 85 in this example).

6 Ranking Recognition Results

6.1 Requirements

In the final stage of the algorithm, pairs of hypotheses whose pose uncertainty regions have a non-empty intersection are ranked by probability. For a ranking scheme to be useful it should exhibit the following characteristics:

- Using the notion of "maximum likelihood interpretations" discussed in [30], more likely interpretations (hypotheses with larger pose uncertainty regions) should be ranked higher than less likely ones.

- Interpretations which would assume a larger uncertainty in vertex position should be ranked lower than interpretations with smaller uncertainty.

- Feature combinations with many plausible interpretations (e.g., features belonging to a single rectangular or triangular face) should be ranked lower than feature combinations with a unique interpretation.

Our probabilistic expression accounts for all these sometime conflicting requirements in ranking possible interpretations. In addition, the algorithm should be able to rank the correct hypotheses first even if the algorithm has to be stopped for lack of time before all pairs of hypotheses have been tested.

6.2 Derivation

Given a set of image features which participate in a match hypothesis (ratio/angle or angle/angle), the pose uncertainty region bounds the region in the pose space in which the error is bounded by ϵ. The higher the value of ϵ the higher the probability that if the hypothesis is correct, that the pose of the object lies within the uncertainty region. ϵ is set large enough such that the probability that the correct pose is not within the uncertainty region is very small.

Given two feature sets in the image e_1 and e_2 and two respective hypotheses h_1 and h_2, we define H as the hypothesis that h_1 and h_2 are true and both match image feature sets to the same instance of a certain model M. We compute $P(h_1, h_2, H, M|e_1, e_2)$, using Bayes' rule:

$$P(h_1, h_2, H, M|e_1, e_2) = \frac{P(e_1, e_2, h_1, h_2, H, M)}{P(e_1, e_2)}. \tag{10}$$

For e_1 and e_2 to be features of the same object, the pose of the object p must be in the intersection of the pose uncertainty regions of the two hypotheses which we denote

by $U(e_1, h_1)$ and $U(e_2, h_2)$ respectively. For each possible pose we write

$$P(e_1, e_2, h_1, h_2, H, M, p) = P(M)P_d(e_1, h_1)P_d(e_2, h_2)\chi_{U(e_1,h_1)}(p)\chi_{U(e_2,h_2)}(p)f_p(p),$$

where $\chi_{U(e_1,h_1)}(p)$ and $\chi_{U(e_2,h_2)}(p)$ are the characteristic functions of $U(e_1, h_1)$ and $U(e_2, h_2)$ respectively and $f_p(p)$ is the p.d.f. of poses in the pose space. If the position of the camera with respect to the surface on which the objects are placed is known, information about stable poses of the objects can be reflected in $f_p(p)$. By marginalizing with respect to p we obtain

$$P(e_1, e_2, h_1, h_2, H, M) = P(M)P_d(e_1, h_1)P_d(e_2, h_2) \int_p \chi_{U(e_1,h_1)}(p)\chi_{U(e_2,h_2)}(p)f_p(p)dp. \tag{11}$$

When there is no information about the pose of the camera we have to assume that poses are uniformly distributed and that the volume of the pose space is normalized to 1. In this case (11) simplifies to:

$$P(e_1, e_2, h_1, h_2, H, M) = P(M)P_d(e_1, h_1)P_d(e_2, h_2)|U(e_1, h_1) \cap U(e_2, h_2)|. \tag{12}$$

We compute $P(e_1, e_2)$ by summing over every pair of hypotheses h_i, h_j which could generate e_1 and e_2 respectively and whether e_1 and e_2 belong to the same object ($H^{(i,j)}$) or not ($\neg H^{(i,j)}$), yielding:

$$\begin{aligned} P(e_1, e_2) = &\textstyle\sum_i \sum_j P(e_1, e_2, h_i, M^{(h_i)}, h_j, M^{(h_j)}, H^{(i,j)}) + \\ &\textstyle\sum_i \sum_j P(e_1, e_2, h_i, M^{(h_i)}, h_j, M^{(h_j)}, \neg H^{(i,j)}), \end{aligned}$$

where $M^{(h_i)}$ is the model to which the model features of h_i belong. In the first term e_1 and e_2 are feature sets of the same instance of a certain model, so the probabilities are computed in the same way that as the numerator of (10). In the second term, e_1 and e_2 do not belong to the same object, therefore

$$P(e_1, e_2, h_i, M^{(h_i)}, h_j, M^{(h_j)}, \neg H^{(i,j)}) = P(e_1, h_i, M^{(h_i)})P(e_2, h_j, M^{(h_j)})P(\neg H^{(i,j)}).$$

$P(\neg H^{(i,j)})$ is computed empirically from typical scenes as the probability that $M^{(h_i)}$ and $M^{(h_j)}$ will both appear in an image. We use the same arguments that were used to derive (11) to compute $P(e_1, h_i, M^{(h_i)})$, by taking the volume of one uncertainty region instead of the intersection of two regions, yielding:

$$P(e_1, h_i, M^{(h_i)}) = P(M^{(h_i)}) P_d(e_1, h_i) |U(e_1, h_i)|. \tag{13}$$

Combining all these results we are able to compute $P(h_1, h_2, H, M | e_1, e_2)$.

This derivation can be easily extended to more than two hypotheses. In the numerator of (10) we compute the intersection of the uncertainty regions of all the hypotheses and in the denominator we sum over all possible interpretations of the feature sets under consideration (all the features belong to the same object, some of them belong to one others to another etc...). We use the results for all subsets of the set of features in order to compute that expression. It is important to note that only sets of features that all of their subsets have non-zero rank might have a non-zero rank themselves. Therefore, we only consider the small number of pairs of hypotheses which have been found by the pair-ranking procedure as input for the extended procedure, which can be performed at minimal computational cost but have very statistically significant results.

During recognition, for every pair of hypotheses whose pose uncertainty regions intersect we evaluate (10). Terms similar to $P(e_1, e_2, h_i, h_j, M, H^{(i,j)})$ appear in the numerator and the denominator of the expression. For hypothesis pairs whose pose uncertainty region do not intersect, this term will be zero. Therefore, when we compute the rank of a hypothesis pair we can assume at first that all the terms of that type except $P(e_1, e_2, h_1, h_2, H)$ are zero. When we compute the rank of another hypothesis pair h_1' and h_2' which interpret the same image feature sets, we will add the value we computed for $P(e_1, e_2, h_1', h_2', M', H)$ to the denominator of the rank of h_1 and h_2. Terms of the type $P(e_1, h_i, M^{(h_i)})$ which also appear in the denominator only involve one hy-

pothesis, therefore their value can be pre-computed and stored in the look-up tables. However that is not always necessary because as the following calculation will show, their values may be very small and their impact on the value of (10) is negligible. We determine whether to neglect these terms by analyzing the relative sizes of terms of the type $P(e_1, e_2, h_i, h_j, M, H^{(i,j)})$ and $\sum_{i,j} P(e_1, e_2, h_i, M^{(h_i)}, h_j, M^{(h_j)}, \neg H^{(i,j)})$ and when the former is much bigger than the latter, the latter can be discarded.

We estimate $|U(e_1, h_1) \cap U(e_2, h_2)|$, in order to estimate the value of $P(e_1, e_2, h_i, h_j, M, H^{(i,j)})$. Consider the case of the one dimensional pose space and that $U(e_1, h_1)$ and $U(e_2, h_2)$ are segments of length l which overlap. When the relative positions of $U(e_1, h_1)$ and $U(e_2, h_2)$ are uniformly distributed, the average length of the overlap between them will be $l/2$. Generalizing this to the six-dimensional pose space, we estimate that $|U(e_1, h_1) \cap U(e_2, h_2)| \approx |U|/2^6$, where $|U|$ is the average volume of a pose uncertainty region. Consider the recognition system with a database of m models. Each model has on average n feature sets, and on average k of them appear in a given scene. We can estimate $P(M) \approx k/m$. Thus we can estimate that

$$P(e_1, e_2, h_i, h_j, M, H^{(i,j)}) \approx 2^{-6}|U|(k/m)P_d(e, h)^2,$$

and

$$\sum_{i,j} P(e_1, e_2, h_i, M^{(h_i)}, h_j, M^{(h_j)}, \neg H^{(i,j)}) \approx (\frac{k}{m})^2(nm)^2|U|^2 P_d(e, h)^2 = k^2 n^2 |U|^2 P_d(e, h)^2$$

The ratio between these two values yields:

$$\frac{P(e_1, e_2, h_i, h_j, M, H^{(i,j)})}{\sum_{i,j} P(e_1, e_2, h_i, M^{(h_i)}, h_j, M^{(h_j)}, \neg H^{(i,j)})} \approx \frac{2^{-6}|U|(k/m)P_d(e, h)^2}{k^2 n^2 |U|^2 P_d(e, h)^2} \approx \frac{1}{2^6 k n^2 m |U|}. \quad (14)$$

Evaluating (14) for a recognition system such that $k \approx 1$,$n \approx 10$ and $|U| \approx 10^{-7}$, gives $1500/m$. Only when the number of models in the database $m > 100$ will the contribution

of $\sum_{i,j} P(e_1, e_2, h_i, M^{(h_i)}, h_j, M^{(h_j)}, \neg H^{(i,j)})$ to (10) be significant, for a smaller database this term can be neglected.

The recognition algorithm traverses the list of the pairs of hypotheses in decreasing probability order. We compute the rank of pairs of hypotheses whose pose uncertainty regions have a non-empty intersection by first evaluating (12) and dividing it by the sum of all the values of (12) computed for all the pairs of hypotheses found so far which suggest interpretations to the same pair of features.

6.3 Characteristics of the Ranking Scheme

In section 6.1, we made several requirements of our ranking scheme. Here we will analyze the algorithm to see how it satisfies these requirements.

We have required that "popular" features which yield many possible interpretations (e.g., features which belong to a rectangular or triangular face) be ranked lower than features which yield few interpretations. "Popular" features will participate in many hypothesis sets. Therefore their corresponding value of $P(e_1, e_2, h_1, h_2, H, M)$ will contribute to the denominators of the probabilities of all the interpretations, thus reducing the ranks of them all. This is reasonable because this set of features does not allow us to discriminate between the different hypotheses, where as a less "popular" but probably correct set of features will have a higher rank, since not many competing hypotheses will exist for that set of features.

We have required that even if the algorithm had to be stopped without checking all pairs of hypotheses, the correct recognition results would be ranked high on the list. As the hypotheses are traversed in decreasing probability order there is a high probability that the correct hypothesis pairs will be ordered high on the list. Therefore, if the recognition process has to be interrupted, we can still assume that the match hypotheses corresponding to the correct interpretation have been processed. We are especially

interested in the "non-popular" feature sets. For them the correct interpretation has been found and there is a small chance that any other competing interpretations would have been found even if all pairs of hypotheses had been checked. Therefore most "non-popular" hypotheses will have a high rank and that rank would be equal to the final rank in many cases.

We also made several requirements on how competing interpretations for the same set of features should be ranked. We demonstrate the performance of the ranking algorithm using illustrations of pose uncertainty regions of typical hypothesis pairs showed in Figure 10.

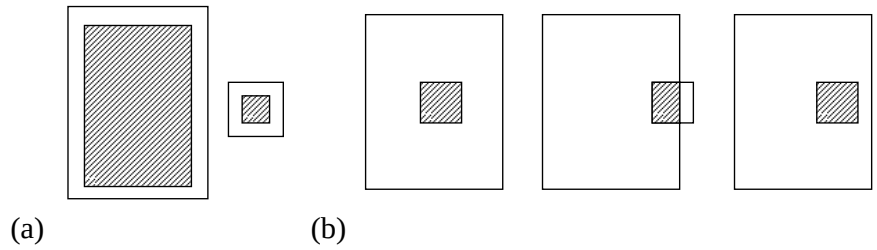


Figure 10: Illustration of the performance of the ranking scheme. Each uncertainty region of an hypothesis is illustrated by a rectangle, their intersection is illustrated by a shaded rectangle. (a) the algorithm ranks higher a pair of likely hypotheses with large pose uncertainty regions (left) than a pair of unlikely hypotheses with small pose uncertainty regions (right); (b) the algorithm ranks higher a hypothesis pair with a small error in vertex position (left) over a pair with a large error (middle) but can not prefer the pair on the right to the pair on the left even though the error for that pair is larger.

The algorithm ranks competing interpretations for a pair of features by comparing the intersection of the pose uncertainty regions of the two pairs of hypotheses. In Figure 10(a), a pair of likely hypotheses with large pose uncertainty regions (left) are ranked higher than a pair of unlikely hypotheses with small pose uncertainty regions(right). Thus the algorithm would prefer the "maximum likelihood" interpretation [30] over the less likely interpretation. In Figure 10(b), we study the case in which the size of the

pose uncertainty regions is the same but their relative positions are different. The closer the centers of the regions are, the smaller the error in vertex position will be if the interpretation is correct. The algorithm ranks higher the hypotheses pair with a small error in vertex position(left) over a pair with a large error(middle) which causes their uncertainty regions to not fully intersect. The algorithm however does not rank the pair on the left higher than the pair on the right even though the pair on the right assumes a larger uncertainty in vertex position because the size of the intersection of the pose uncertainty regions is the same. In the next section we present a variant of the ranking scheme which addresses this problem.

6.4 Exact Ranking Scheme

The fundamental characteristic of the algorithm which prevents it from discriminating between the two interpretations illustrated in Figure 10(b)(left,right) is that all poses within the intersection of the pose uncertainty regions have equal weight even though the poses which yield small vertex position errors should have a higher weight than poses which yield large errors.

To solve this problem we will weight each pose by the distance between the image features and the model features back-projected using that pose. Assuming the uncertainty in vertex position has a normal (or any other known) distribution, we use the probability density function value for the computed vertex position uncertainty as the weight for that pose. Substituting this expression into (11) yields:

$$P(e_1, e_2, h_1, h_2, H, M) = P(M)P_d(e_1, h_1)P_d(e_2, h_2) \int_p f_1(p) f_2(p) f_p(p) dp, \qquad (15)$$

where $f_1(p)$ and $f_2(p)$ denote the probability density functions applied to the errors in e_1 and e_2 respectively assuming the pose is p.

Similarly (13) is transformed into:

$$P(e_1, h_i, M^{(h_i)}) = P(M^{(h_i)}) P_d(e_1, h_i) \int_p f_1(p) f_p(p) dp.$$

This ranking scheme correctly ranks 10(b)(left) higher than 10(b)(right). In order to use this scheme we would have to evaluate expressions of the type (15) during recognition time. There is no closed form solution for evaluating integrals of that type and numerical Monte-Carlo integration techniques must be used. These techniques are computationally very costly so we recommend using the simpler recognition ranking scheme presented in Section 6.2.

7 Experimental Recognition Results

In this section we present the implementation of our recognition algorithm and show experimental results of running it on real images. Our model database consists of the seven objects shown in Figure 11.

We extract edges from the image using the Canny edge detector [10] and then detect lines from the extracted edges. We combine these lines automatically into feature sets using the following technique: we detect corners in the image as the intersection point of the supporting lines of two lines in the image when the actual termination points of the two lines is close to the intersection point. Once a corner has been detected, additional lines which terminate close to the corner are added to the list of lines emanating from it. We label lines which end in the middle of another line (T junctions) as partially occluded edges. We generate feature sets from this information. For each line triple emanating from a corner, we generate an angle/angle feature set. For each pair of lines emanating from a corner, we generate an ratio/angle feature set when both lines start and end at a vertex, or an occluded ratio/angle feature set when one or both of them ends at a T

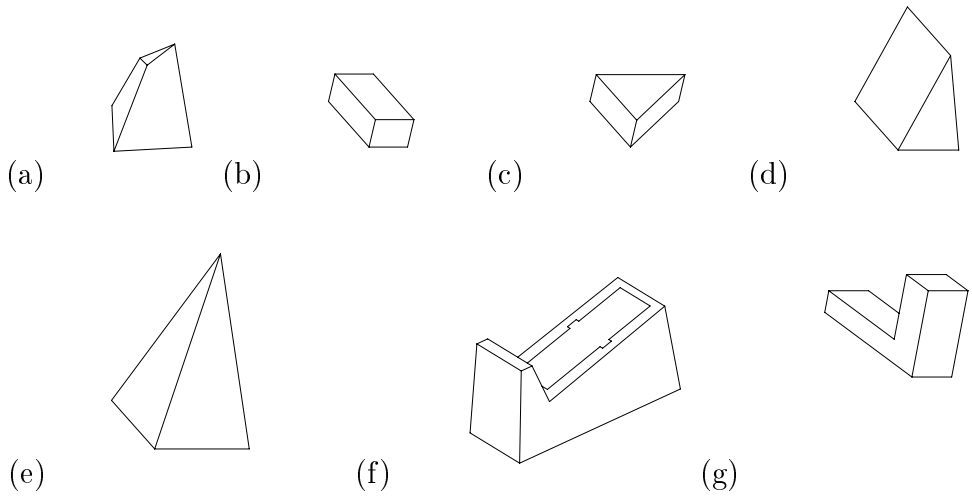


Figure 11: Model database: (a) a truncated pyramid, (b) a box, (c) a triangular prism, (d) another prism, (e) a pyramid, (f) a tape dispenser, (g) an L shaped object.

junction. The feature sets extracted in this stage need not be perfect and not all visible features sets must be found because an important feature of our recognition algorithm is that it is robust to uncertain and incomplete input.

For each feature set, we retrieve from the look-up tables the corresponding match hypotheses and their probabilities. Occlusion is accounted for in matching the feature sets with the match hypotheses. A list of pairs of hypotheses is generated. The a-priori probability estimate for each such pair is set as the product of probabilities of the two hypotheses which has been retrieved from the look-up table. This list is then sorted by probability and processed in that order. For each hypothesis pair, we test if their pose uncertainty regions intersect. We compute the rank of each compatible pair of matches found and maintain a list of compatible pairs of matches sorted by rank. The algorithm outputs the interpretations due to the pairs of matches with the highest ranks. As was

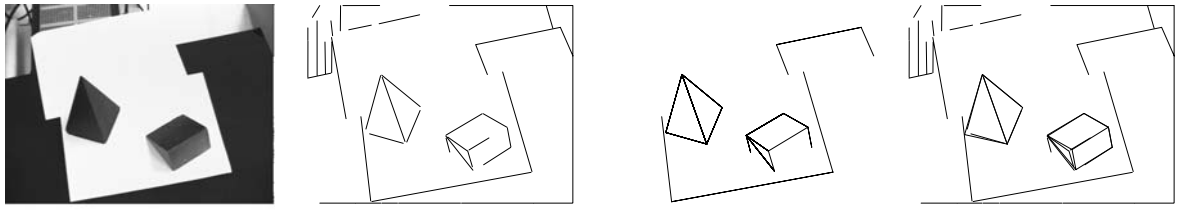
explained in Section 6.2, this ranking scheme can be extended to deal will larger sets of features with a small extra computational cost. In this implementation however, only pairs of matches were considered.

To make this algorithm a complete recognition system the following two steps have to be added: least squares estimation of pose, and hypothesis verification by back-projection. These two steps have not been implemented since the focus was put on finding efficient ways to generate promising hypotheses.

We now present several examples of results obtained running our algorithm on real images. Figure 12(a) shows an image of the rectangular pyramid and the second prism. Note that the results of the edge detection and line extraction (Figure 12(b)) contain features that are due to the background and shadows. Figure 12(c) shows the features belonging to feature sets extracted from the image. Note that the line due to the shadow of the prism is part of angle/angle feature sets where the other lines participating in those feature sets are edges of the prism itself. Edges which have not been fully extracted by the edge detector may only participate in angle/angle feature sets and are discarded when they are not adjacent to a trihedral corner. Applying this criterion to this image caused most features due to the background to be discarded. The objects recognized by the algorithm are shown in Figure 12(d).

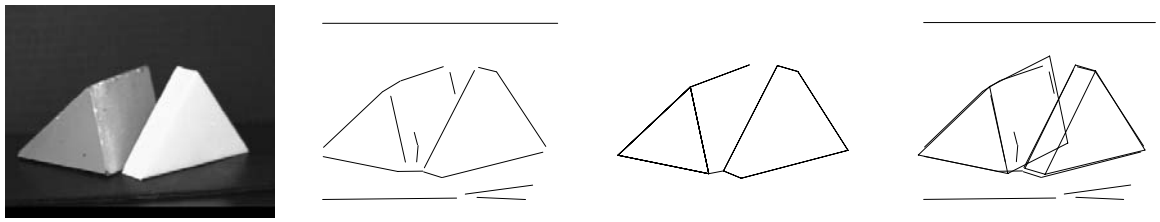
Results of running the algorithm on an image of two triangular prisms with partial occlusion are shown in Figure 13. Note that the edge detector was not able to recover the internal edge of the triangular shape but found all the silhouette edges.

In Figure 14, the results of processing another image of the two triangular prisms with partial occlusion are shown. Here, a different part of the second prism is occluded. Note that again internal edges were not detected by the edge detector. The internal edge of the second prism is especially interesting. Parts of the edge were detected but not the whole edge. Therefore the segments of the edge participate only in angle/angle feature



(a) (b) (c) (d)

Figure 12: Recognition results for an image of a rectangular pyramid and a prism: (a) the image; (b) the results of edge and line detection; (c) feature sets recovered from the image; (d) recognized objects.

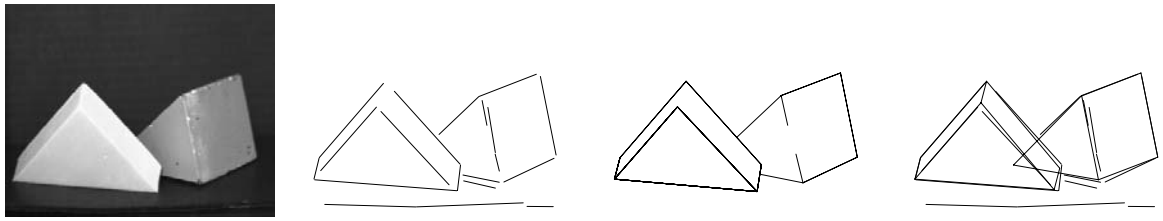


(a) (b) (c) (d)

Figure 13: Recognition results for an image of two triangular prisms with partial occlusion: (a) the image; (b) the results of edge and line detection; (c) feature sets recovered from the image; (d) recognized objects.

sets but not in ratio/angle feature sets.

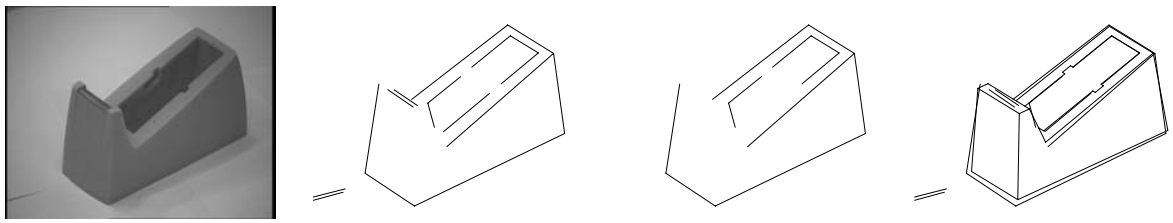
In Figure 15, the results of processing an image of the tape dispenser are shown. Note that the the object is not purely polyhedral but even so the program was able to recognize the object. As usual only some of the visible features were detected by the edge/line detector. One of the ratio/angle feature sets is partially occluded. Naturally it was matched only to model feature sets where the corresponding edge was partially occluded. As these hypotheses are quite rare in our database, the correct hypothesis got a very high score and participated in the first hypothesis pair that was found and the



(a) (b) (c) (d)

Figure 14: Recognition results for another image of two triangular prisms with partial occlusion: (a) the image; (b) the results of edge and line detection; (c) feature sets recovered from the image; (d) recognized objects.

hypothesis pair was correct.



(a) (b) (c) (d)

Figure 15: Recognition results for an image of the tape dispenser: (a) the image; (b) the results of edge and line detection; (c) feature sets recovered from the image; (d) recognized object.

We collected in Table 2 statistics regarding the run of the algorithm on the examples shown above. For each run, we tabulated the number of feature sets extracted from the image, the number of match hypotheses retrieved from the look-up tables, the number of pairs of match hypotheses which were tested for compatibility, the number of compatible pairs found, how many of them were correct and how many of the first ten compatible pairs ranked by probability were actually correct and their ranks.

In Table 3, we display the running times of the algorithm on the four images. For each run we tabulated the time it took until all the hypothesis pairs were generated and

sorted, the time it took the algorithm until a correct hypothesis pair of one of the two objects has been tested and found to be correct, the time until a correct hypothesis pair belonging to the other object is tested and the total run time of the algorithm which terminates after testing all hypothesis pairs. The algorithm was implemented in C++ and run on a Sun E450 server.

It is important to note that due to the rank of the match hypotheses most of the correct results will be found early in the run of the algorithm and there is no need to test all possible hypothesis pairs. Also note that our main concern was to explore the concepts underlying these algorithms, and little effort was made to implement this algorithm efficiently.

Figure	Feature Sets	Match Hypotheses	Hypothesis Pairs	Possible Results	Correct Results	Correct Results of the 10 Highest Ranked Results
12	25	4865	667904	877	192	3 (5,9,10)
13	11	2221	100076	318	25	2 (8,9)
14	16	3117	206575	627	58	4 (3,4,6,8)
15	11	2278	110265	135	23	7 (1-3,4-7,9)

Table 2: The recognition results table shows statistics from various stages of the run of the recognition algorithm.

Figure	Time Until Hypothesis Pairs are Generated	Time Until First Object Hypothesis	Time Until Second Object Hypothesis	Total Run Time
12	15 sec	17 sec	57 sec	177 sec
13	2 sec	4 sec	11 sec	22 sec
14	5 sec	6 sec	10 sec	86 sec
15	3 sec	3 sec	N/A	11 sec

Table 3: Timing of the run of the algorithm: time until the hypothesis pairs are generated; time until the first hypothesis pair which correctly recognizes the first object was tested; time until the first hypothesis pair which correctly recognizes the second object was tested; time until the algorithm completed.

We show several incorrect hypothesis pairs found by our algorithm in Figure 16 in order to characterize them and suggest methods to avoid processing the pairs of matches which yielded them in the first place. In Figure 16(a), the interpretation of the scene is correct, only the pose of the object is wrong. However, as the internal edges of the object were not found by the edge detector there is no way to distinguish between the correct and incorrect pose, they both yield the same silhouette. A similar example is shown in Figure 16(b). However in this case internal edges were recovered by the edge detector and the correct match should be ranked higher when more features are added to the match.

The examples in Figures 16(c,d) show how any rectangular or triangular face can be matched to any other face of the same type. In order to avoid this type of erroneous pairs of matches, it is better not to process pairs of hypotheses whose model features belong to the same face at all. Although correct hypothesis pairs will also be discarded, the performance of the recognition algorithm will not be hurt because the algorithm could not distinguish between the correct pair and the many incorrect ones and gives them all a low rank.

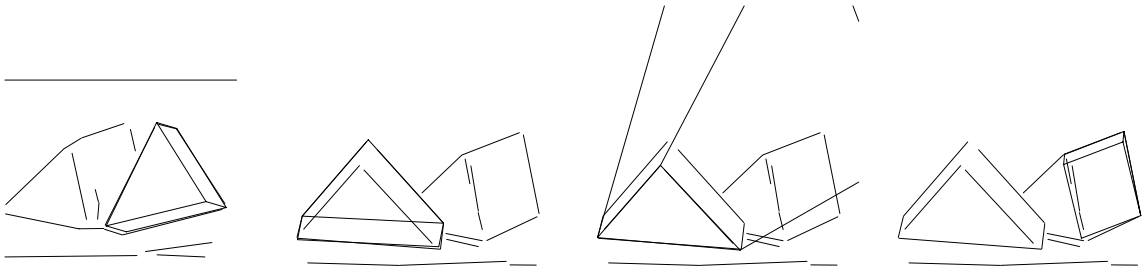


Figure 16: Wrong recognition results: (a) the triangular object with an incorrect pose; (b) the triangular object with another incorrect pose which contradicts some of the other features extracted from the image; (c) a triangular face of the triangular object is matched to a face of the truncated pyramid; (d) a rectangular face of the prism is matched to the wrong rectangular face of the prism.

We therefore re-ran the algorithm discarding all pairs of hypotheses which lie on the same face and all hypotheses which have edges in common. The results are presented in Tables 4 and 5. The number of hypothesis pairs has been reduced somewhat, and the running time has also improved. The major difference is the reduction by a factor of 10-80 of the number of recognition results found. Most of the remaining results left are correct. As a result there are more correct results in the top 10 results. Therefore, not only will the total number of results which might have to be verified is reduced, a correct answer will be one of the first tested.

In order to test the effects of clutter on our algorithm, we re-ran the program on Figure 15 to which we added 10 and even 30 random feature sets. In both cases the program ran longer but not even one additional false result was found. This demonstrates the fact that problem of incorrect results is usually not due to clutter but is due to similarity between models.

Figure	Feature Sets	Match Hypotheses	Hypothesis Pairs	Possible Results	Correct Results	Correct Results of the 10 Highest Ranked Results
12	25	4865	620181	69	54	6 (4-5,6-10)
13	11	2221	92966	4	4	4 (1-4)
14	16	3117	189204	24	12	5 (2,5,8-10)
15	11	2278	101842	10	9	9 (1-8,10)

Table 4: The recognition results table shows statistics from various stages of the second run of the recognition algorithm.

8 Discussion

In this paper we have presented a probabilistic 3D object recognition algorithm. We have studied the nature of the iso-ratio and iso-angle curves and traced them on the viewing sphere. We have used these curves to accurately compute conditional probabilities that

Figure	Time Until Hypothesis Pairs are Generated	Time Until First Object Hypothesis	Time Until Second Object Hypothesis	Total Run Time
12	18 sec	22 sec	41 sec	134 sec
13	3 sec	10 sec	11 sec	21 sec
14	6 sec	10 sec	28 sec	63 sec
15	3 sec	3 sec	N/A	10 sec

Table 5: Timing of the run of the algorithm with "popular" feature set pairs removed: time until the hypothesis pairs are generated; time until the first hypothesis pair which correctly recognizes the first object was tested; time until the first hypothesis pair which correctly recognizes the second object was tested; time until the algorithm completed.

image features match model features. The probabilities have been incorporated into a probabilistic model which takes into account the uncertainties inherent in the input to the recognition algorithm. These probabilities have been used to decide the order in which to process the match hypotheses. We have developed a method to compute the pose of the object using a minimal feature set in the image which matches to feature set in the model database. Taking the uncertainty in values measured in the image into account, we have computed the uncertainty pose region for each hypothesis. By finding hypotheses whose pose regions have a non-empty intersection, we have been able to find a set of feature sets which reinforce the recognition hypothesis. We have ranked these hypotheses by computing the probability that all the feature sets in the set came from the same instance of the suggested object.

Future work will be dedicated to improving the efficiency of the algorithm by finding better ways to order the rank hypotheses, process hypothesis pairs and find smaller and more accurate pose uncertainty regions. Another important research direction would be to extend this probabilistic recognition scheme to deal with more complicated objects such as curved objects.

Acknowledgments

This work was supported in part by the Beckman Institute and the Center for Advanced Study of the University of Illinois at Urbana-Champaign, by the National Science Foundation under grant IRI-9224815, and by the National Aeronautics and Space Administration under grant NAG 1-613.

References

[1] T. D. Alter. 3D pose from 3 points using weak-perspective. *IEEE Trans. Patt. Anal. Mach. Intell.*, 16(8):802–808, August 1994.

[2] T. D. Alter and D. W. Jacobs. Error propagation in full 3D-from-2D object recognition. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, pages 892–898, Seattle, Washington, June 1994.

[3] R. D. Arnold and T. O. Binford. Geometric constraints in stereo vision. In *Proc. SPIE meeting*, San Diego, California, July 1980.

[4] N. Ayache and O. D. Faugeras. HYPER: a new approach for the recognition and positioning of 2D objects. *IEEE Trans. Patt. Anal. Mach. Intell.*, 8(1):44–54, January 1986.

[5] J. Ben-Arie. The probabilistic peaking effect of viewed angles and distances with application to 3-D object recognition. *IEEE Trans. Patt. Anal. Mach. Intell.*, 12(8):760–774, August 1990.

[6] P.J. Besl and R.C. Jain. Three-dimensional object recognition. *ACM Computing Surveys*, 17(1):75–145, march 1985.

[7] T.O. Binford, T. Levitt, and W. Mann. Bayesian inference in model-based machine vision. In *Workshop on Uncertainty in Artificial Intelligence*, 1987.

[8] T.M. Breuel. Fast recognition using adaptive subdivision of transformation space. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, pages 445–451, 1992.

[9] J. B. Burns, R. S. Weiss, and E. M. Riseman. View variation of point-set and line-segment features. *IEEE Trans. Patt. Anal. Mach. Intell.*, 15(1):51–68, January 1993.

[10] J.F. Canny. A computational approach to edge detection. *IEEE Trans. Patt. Anal. Mach. Intell.*, 8(6):679–698, November 1986.

[11] T.A. Cass. Robust geometric matching for 3d object recognition. In *ICPR*, pages A:477–482, 1994.

[12] R.T Chin and C.R. Dyer. Model based recognition in robot vision. *ACM Computing Surveys*, 18(1):67–108, January 1986.

[13] O.D. Faugeras and M. Hebert. The representation, recognition, and locating of 3-D objects. *International Journal of Robotics Research*, 5(3):27–52, Fall 1986.

[14] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24(6):381–395, June 1981.

[15] P.C. Gaston and T. Lozano-Pérez. Tactile recognition and localization using object models: The case of polyhedra in the plane. *IEEE Trans. Patt. Anal. Mach. Intell.*, 6(3), 1984.

[16] W. E. L. Grimson and D. P. Huttenlocher. On the verification of hypothesized matches in model-based recognition. *IEEE Trans. Patt. Anal. Mach. Intell.*, 13(12):1201–1213, December 1991.

[17] W. E. L. Grimson, D. P. Huttenlocher, and T. D. Alter. Recognizing 3D objects from 2D images; an error analysis. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, pages 316–321, Champaign, Illinois, June 1992.

[18] W.E.L. Grimson and T. Lozano-Pérez. Model-based recognition and localization from sparse range or tactile data. *International Journal of Robotics Research*, 3(3), 1984.

[19] R. M. Haralick, C. Lee, K. Ottenberg, and M. Nölle. Review and analysis of solutions of the three point perspective. *Int. J. of Comp. Vision*, 13(3):331–356, December 1994.

[20] J. Hornegger and H. Niemann. Statistical learning, localization, and identification of objects. In *ICCV*, pages 914–919, 1995.

[21] D. Huttenlocher and S. Ullman. Recognizing 3D solid objects by alignment with an image. *Int. J. of Comp. Vision*, 5(2):195–212, 1990.

[22] D.J. Kriegman. Let them fall where they may: Capture regions of curved objects and polyhedra. *International Journal of Robotics Research*, 16(4):448–472, August 1997.

[23] D.J. Kriegman and J. Ponce. A new curve tracing algorithm and some applications. In P.J. Laurent, A. Le Méhauté, and L.L. Schumaker, editors, *Curves and Surfaces*, pages 267–270. Academic Press, New York, 1991.

[24] D. Lowe. *Perceptual Organization and Visual Recognition.* Kluwer Academic Publishers, Boston, 1985.

[25] A.P. Morgan. *Solving Polynomial Systems using Continuation for Engineering and Scientific Problems.* Prentice Hall, Englewood Cliffs, NJ, 1987.

[26] C. F. Olsen. Fast alignment using probabilistic indexing. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, pages 387–392, New York, New York, June 1993.

[27] C. F. Olsen. Probabilistic indexing for object recognition. *IEEE Trans. Patt. Anal. Mach. Intell.*, 17(5):518–521, May 1995.

[28] W. Rucklidge. Locating objects using the hausdorff distance. In *ICCV*, pages 457–464, 1995.

[29] I. Shimshoni. On estimating the uncertainty in the location of image points in 3d recognition from match sets of different sizes. *Comp. Vis. Im. Understanding*, 74(3):163–173, June 1999.

[30] D. Weinshall, M. Werman, and N. Tishby. Stability and likelihood of views of three dimensional objects. In *Proc. European Conf. Comp. Vision*, pages 24–35, Stockholdm, Sweden, June 1994.

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

[32] Y. Wu, S. S. Iyenger, R. Jain, and S. Bose. A new generalized computational framework for finding object orientation using perspective trihedral angle constraint. *IEEE Trans. Patt. Anal. Mach. Intell.*, 16(10):961–975, October 1994.