Uncertainty Reasoning in Object Recognition by Image Processing

Robert Hummel

Courant Institute, New York University

Abstract

Object recognition in digital images is a primary issue in robotics. We consider the model-based vision problem, where objects to be recognized come from a database of geometrically precise models. However, the modeling process involves uncertainties, and thus predicted collections of features will be subject to possible variations. Likewise, the image analysis problem using digital images must deal with sensor noise and ambiguity in the imaging process. Accordingly, object recognition is not a simple matter of matching features sets, but must deal with variabilities in the models and in the extracted features in the scenes. In this paper, we consider how these uncertainties should be handled. We describe how predicted variability can be used to compute a match metric, in order to assess the quality of possible models. We discuss two methods for dealing with extracted uncertainty. Finally, we speculate on other methods of assessing uncertainty in the recognition process.

1 Introduction

There is a large and active field called uncertainty reasoning, which includes uncertainty calculi, such as the Dempster/Shafer theory of evidence, fuzzy logic, and other methods (e.g., see [1]). However, our interest here is with the recognition of objects in digital images, and is related to pattern recognition and model-based vision. The question naturally arises: How can uncertainty calculi be used in order to perform uncertainty reasoning so as to benefit the task of object recognition? In this paper, we wish to go back to basic principles, to discuss the object recognition problem from the standpoint of uncertainty reasoning. Rather than making the problem fit the calculus, this paper ponders the derivation of a calculus to fit the problem. It is possible that Bayesian nets, Kalman filters, and other uncertainty management schemes have direct applications to object recognition, but this is not our approach here.

There are accordingly many kinds of uncertainty that we will substantially ignore. When we speak of uncertainty, we often mean something orthogonal to probability. For example, one can say that there is a 50% probability of rain, and be very certain of the prediction. On the other hand, one might say that there is a 50% probability of rain because one has no evidence, and is simply giving a prior probability. In this case, one could say that the degree of certainty is very low. The degree of certainty could also be low because the information that is known gives conflicting clues. (See [2] for considerations of such notions of uncertainty.) In performing object recognition, we will take an hypothesis H that a particular object is present in a particular orientation in the image, and evaluate the probability that the hypothesis is true given the information in the image. Measures of the degree of certainty in this probability evaluation are problematic. We will consider a couple potential measures of certainty, and consider briefly how such measures could be used, but our first order of business is to determine formulas for evaluating the probability.

2 Object recognition as feature matching

We begin by considering the object recognition problem. We are interested in recognizing objects in digital images, where the objects come from a library of models. So called model-based vision is a simpler problem than the more general scene analysis problem, where the objects have abstract models and significant variability. Humans can recognize chairs based on suppositions of the functionality of an object. For automatic object recognition systems, for the time being, we would be glad to have a system that can discern objects for which there is a rather precise advance model.

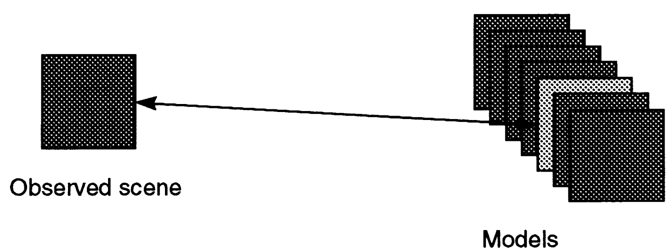

Figure 1. Matching models.

Accordingly, we have an observed scene, and a potentially large collection of models (Figure 1). We wish to find a match between a portion of the observed scene and one of the models, and often the match should be invariant to translation, rotation, or other geometric transformations. The system designer must decide on the transformation class under which recognition should be invariant, and determine a method for representing the information in the image so as to facilitate the match. Finally, there is the issue of how to efficiently perform the matching function, given the precise formulation.

In this work, we consider objects to be represented by a collection of features. A feature is simply a construct extracted from the image, and represented by a vector of values that describe attributes of the feature. In digital image processing, most features will have a location (x,y), but most features will also have other attributes, such as an orientation. Figure 2 shows some examples of features in images that we

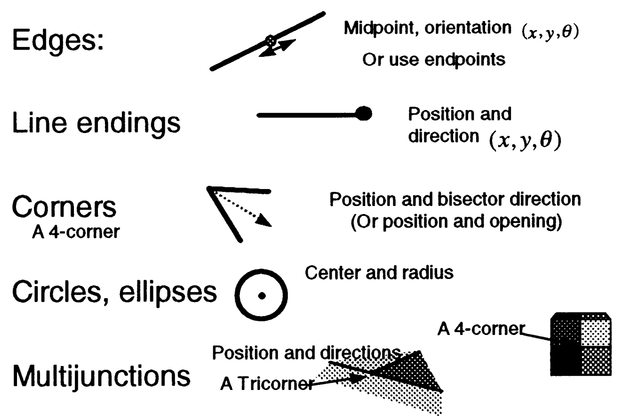

Figure 2. Types of features in images.

consider relevant, although there may be other kinds of features. The development of discriminating features is a challenging aspect of image analysis research.

Note that each of these features can be represented by a vector of values $(x, y, \theta_1, \cdots)$, where the position is given by (x,y) and the remaining parameters are typically orientation information. However, other feature types might have other types of attributes.

A *pattern* is simply a set of features. When we take an image of a model and extract features from the model, we obtain a pattern. There is no requirement that all the features be of the same type: A pattern can contain edges, corners, multi-corners,

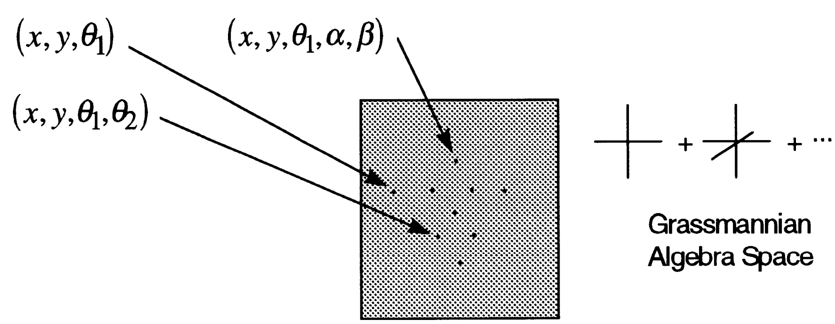

Figure 3. Features have different dimensionalities.

and other feature types. In fact, the features in a pattern can have different dimensionalities. Some of the features might be represented by 3-vectors, some by 4-vectors, and some by 10-vectors. Thus the domain of the features in a pattern consists of the exterior sum of Euclidean spaces of different dimensions, which is the domain of a Grassmannian algebra (see Figure 3). However, describing it as such is a bit of a joke, since we will never use the algebra properties. The point is that the set of features in a pattern is a heterogeneous collection.

With this formulation, we can now view model-based object recognition as the problem of finding matches between patterns of features. We extract a pattern from the scene, and have a database of patterns from a set of models, and we wish to find a match between the extracted pattern and a model pattern. To slightly complicate the situation, we typically want to perform the pattern matching in a translation invariant fashion, and sometimes we additionally want rotation and scale invariance (affine invariance is also possible, but can lead to instability). One way to accomplish translation invariance is to use the notion of *basis sets.* A basis set is a minimum number of attributed features sufficient to define a coordinate system. Then, instead of representing each model as a pattern of features, we multiply represent each model as a set of normalized features, where the features have been transformed with respect to a basis set. Using all possible reasonable basis sets, each model bifurcates into a collection of models/basis-set patterns. That is, for each model, for each (reasonable) basis set, we obtain a normalized pattern of features, which can now be viewed as a single pattern in the database for matching. Likewise, the extracted set of features in the scene explode into a collection of patterns, for each possible basis set in the scene. If we select a single basis set in the scene, then

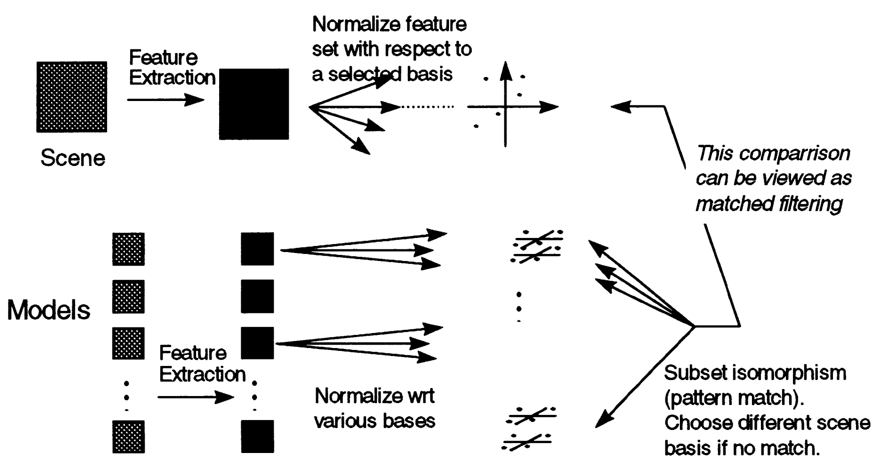

Figure 4. Normalizing patterns with respect to basis sets.

we get a single pattern which, as before, is to be matched against the database of model patterns (which are themselves normalized with respect to basis sets). (See Figure 4.) If no match is found, then a different basis set in the scene should be chosen.

The precise method of performing the matching of patterns is an implementation issue, which we will not consider here. There are matched filtering approaches, branch-and-bound search approaches, and hashing approaches [3,4]. From a computer science standpoint, these implementation issues are interesting and the central concern. However, we have yet to consider uncertainty, and so we next turn our attention to the development of the metric that should be used in finding the match.

3 Features are random vectors

The same image digitizer viewing the same scene will produce different results, depending upon slight variations in the aiming direction of the camera, sensor noise, environmental conditions, and other factors. Accordingly, when we extract features, even if the scene is always the same and the object is always in the same position, we can expect variability. Tracing this variability through to the individual feature vectors, we observe that a feature vector in a pattern is not really a single vector of components, but instead can be represented by a probability distribution over the space of possible vectors. Typically, the distribution will be localized to a nominal vector, but the variability in the extracted feature can be represented by a random vector, i.e., a vector-valued function over a sample space, where probabilities and statistics can be defined. Random vectors are characterized by their probability density function (pdf), which is often modeled by a multivariate Gaussian distribution, but in general can have any form.

Similarly, the prediction of a feature in a pattern should be scene as a random vector. When we say that a model has a predicted pattern of features, we do not mean that each feature vector will necessarily be observed at precisely a given location in Euclidean space. Instead, we mean that there is likely to be a feature found somewhere in some region in the Euclidean space, corresponding to some physical event in the scene.

There is a delicate distinction between the variability that will be observed in the extracted features (from the scene) and the variability in the predicted features. The question is succinctly put as follows: Should the prediction attempt to account for the variability that will occur in the extraction? Prediction has its own sources of variabilities, such as model variations, modeling error, pose uncertainty, etc. Our answer, here, is that the prediction variability should account exclusively for these variations, and not attempt to predict extraction variations. That is, variability in the predicted features occurs exclusively because the prediction is not absolutely certain of its models. The prediction is independent of the sensor that will actually be used to observe the scene (although the type of sensor is known). Extraction of features from the scene will attempt to find precisely the same features that were predicted, and will produce random vectors with probability density functions, where the breadth of the pdf's are influenced by the accuracy of the sensor system.

Also note that the variability in the extracted features is not to be matched to the variability in the predicted features. The variabilities are independent. Regardless of the variabilities in one pattern, the variabilities in the other pattern are ideally point masses.

As an example (see Figure 5), consider an edge feature, containing a location (x,y), and an orientation θ. The location is subject to variability, and is represented by a Gaussian distribution in the location information, and the orientation can also

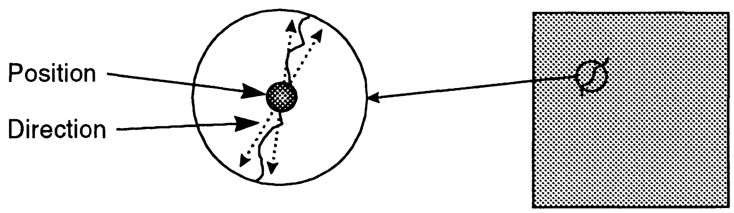

Figure 5. Variability for an edge feature.

vary, which we might posit is supported in a wedge region around a nominal orientation. Depending on the situation, this variability might be called "uncertainty" in the extracted information or the predicted information. We prefer to use the term "variability" because the pdf's might come from precisely modeled information. But the issue of terminology is unimportant compared to the issue of how to use the variability information. We will first make use of the predicted variability.

4 Bayesian Match Metric uses predicted variability

First, some simple Bayesian formulas. If H is any hypothesis, and $E_1, \cdots, E_n$ are conditionally independent pieces of evidence, we have:

$$\Pr(H | E_1, \cdots E_n) = \Pr(H) \cdot \prod \frac{\Pr(H | E_i)}{\Pr(H)}$$

$$= \Pr(H) \cdot \prod \frac{\Pr(E_i | H)}{\Pr(E_i)}.$$

Thus

$$\log\left[\Pr(H | E_1, \cdots E_n)\right] = \log\left[\Pr(H)\right] + \Sigma \log\left[\frac{\Pr(E_i | H)}{\Pr(E_i)}\right].$$

If we have a collection of different hypotheses, $H_k, k = 1, \cdots, N$, maximum likelihood recognition says that we should find the index k that maximizes $\log\left[\Pr\left(H_k | E_1, \cdots, E_n\right)\right]$. By using the monotonic log function, we see that the computation of the maximum probability is the same as maximizing the log probability, which is obtained by maximizing a sum. Each term in the sum is a log likelihood ratio, and must be interpreted for our application domain.

Consider again the feature matching problem, as discussed in Section 2. Let us suppose that H stands for the hypothesis that a pattern of features $\{\mathbf{x}_i\}_{i=1}^{n}$ match, in a one-to-one fashion, with a collection of features $\{\mathbf{y}_i\}_{i=1}^{n}$, and further, that the associations are that $\mathbf{x}_1$ matches $\mathbf{y}_1$, etc. That is, we have re-ordered the features in the scene and the model, to provide n match pairs in order. We will think of the $\mathbf{x}_i$ features as the extracted features, and the $\mathbf{y}_i$ features as the nominal predicted features. Recall, however, that the predicted features have associated variabilities, sometimes called uncertainties, represented by pdf's. Note that since the hypothesis includes an ordering of features, there are many possible hypotheses. Even for the fixed collection of n pairs of features, there are $n!$ possible permutations, and we will want to maximize the log probability over all these possible permutations. But for now, let us consider each associated pair $(\mathbf{x}_i, \mathbf{y}_i)$ as a piece of evidence E_i which should contribute $\log(\Pr(E_i|H)/\Pr(E_i))$ to the total log probability (see Figure 6), assuming independence of the individual pieces of evidence. There may be additional

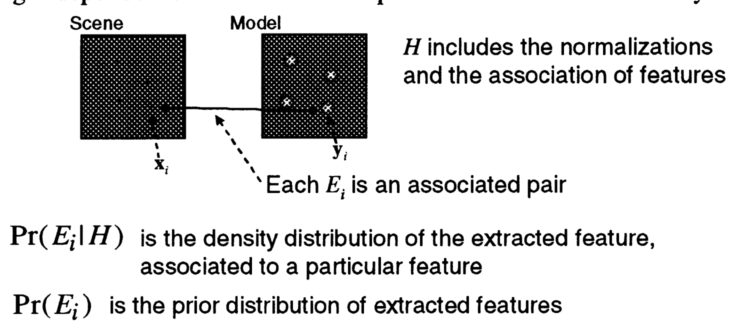

Figure 6. Interpreting the components of the log probability computation.

extracted features, which are unmatched to predicted features, and there may be predicted features that are unmatched to extracted features, but for the moment, we will ignore these sources of evidence.

In this formulation, the evaluation of $\Pr(E_i|H)$ assumes that the predicted feature $\mathbf{y}_i$ matches the extracted feature $\mathbf{x}_i$. The predicted feature $\mathbf{y}_i$ is associated with a distribution function, which we will denote by $f_i(\cdot - \mathbf{y}_i)$. Since the hypothesis posits a match, the probability, which is actually an evaluation of a density function, is simply $f_i(\cdot - \mathbf{y}_i)$. The denominator, $\Pr(E_i)$, is simply the prior probability that extracted feature $\mathbf{y}_i$ appears where it does; the predicted feature $\mathbf{y}_i$ is irrelevant, since there is no hypothesis that the two features match. In fact, the distribution function for $\mathbf{x}_i$ is generally independent of i since there is no reason to distinguish one extracted feature from another extracted feature. (The distribution function might depend on the *type* of the extracted feature, however.) Let us denote the evaluation of the prior distribution function for the extraction of the feature $\mathbf{x}_i$ by the function $\rho(\mathbf{x}_i)$. Thus the total vote due to E_i, computed as $\log(\Pr(E_i|H)/\Pr(E_i))$, is:

$$\log\left(\frac{f_i(\mathbf{x}_i - \mathbf{y}_i)}{\rho(\mathbf{x}_i)}\right),$$

and the total vote for the log probability of the hypothesis $Q = \log(\Pr(H| E_1, \cdots, E_n)$ is:

$$Q = \log(\Pr(H)) + \sum \log\left(\frac{f_i(\mathbf{x}_i - \mathbf{y}_i)}{\rho(\mathbf{x}_i)}\right)$$

We now wish to take into account the fact that not all predicted features will have matches. We will not account for unmatched features in this formulation, but unmatched predicted features can have a large influence on the total score. The difficulty with the formula above is that if one predicted feature $\mathbf{y}_i$ is poorly paired with an extracted feature $\mathbf{x}_i$ that is distant from $\mathbf{y}_i$, then the log term can be a large negative value, and will swamp the sum, effectively denying the hypothesis.

Instead, let us suppose that each predicted feature $\mathbf{y}_i$ is associated with a probability of occurrence, say β_i, and that if it occurs, then the probability density function is $f_i(\cdot - \mathbf{y}_i)$. The important point is that there is a $(1 - \beta_i)$ probability that the feature does not occur at all. This is a predicted probability that the feature will be obscured or otherwise missing.

Then, the hypothesis H includes the following information: (1) That model m is present, (2) that predicted features $\{\mathbf{y}_i\}_{i=1}^{n}$ are present, (3) predicted features $\{\mathbf{y}_i\}_{i=n+1}^{N}$ are not present, and that extracted features $\{\mathbf{x}_i\}_{i=1}^{n}$ match up one-to-one with the predicted features $\{\mathbf{y}_i\}_{i=1}^{n}$ with feature $\mathbf{x}_i$ matching the extracted feature $\mathbf{y}_i$. As before, there are n pieces of evidence, with $(\mathbf{x}_i, \mathbf{y}_i)$ providing the evidence E_i. Further, we still have the background distribution $\rho(\mathbf{x}_i)$ and the predicted feature information $(\beta_i, \mathbf{y}_i, G_{C_i})$, where we have now represented the predicted pdf by a Gaussian function with covariance C_i (see Figure 7).

Incorporating probabilities of non-obscuration in the formulation of the predicted features, and including the nomination of certain predicted features as

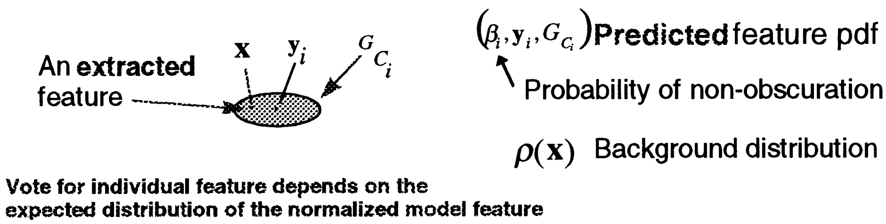

Figure 7. Extracted and predicted pairs.

appearing in the hypothesis, influences the bias probability term $\Pr(H)$, and has no influence on the individual evidence terms. With the hypothesis that the first n predicted features are matched, and the remaining $N-n$ features are not matched, results in the formula for the match value $Q = \log\left(\Pr(H \mid E_1, \cdots, E_n)\right)$:

$$Q = \log\left(\Pr(H_m)\right) + \sum_{i=1}^{n} \log(\beta_i) + \sum_{i=n+1}^{N} \log(1-\beta_i) + \sum_{i=1}^{n} \log\left(\frac{G_{C_i}\left(\mathbf{x}_i - \mathbf{y}_i\right)}{\rho\left(\mathbf{x}_i\right)}\right).$$

In deriving this formula, certain independence assumptions have been necessary, over and above the assumptions that the evidences E_i are independent. In particular, we have decomposed the hypothesis H into a set of independent assumptions, which include the hypothesis H_m that the model m is present, and the hypotheses that the first n predicted features appear in the scene, and the remaining $N-n$ features are not present. Here, we have assumed that each of these components are independent, and so the log probability can be obtained by summing the log probabilities of the individual components. These independence assumptions, however, are less justifiable than the independence of the evidences, since, for example, obscuration of a collection of features might be caused by a single physical object, and thus provides a correlation for the non-presence of a collection of predicted features.

As noted earlier, there are many possible hypotheses, composed of many combinations. Each possible normalized model (which itself involves a pattern and a basis set), and each possible combination of associations of subsets of predicted features to extracted features results in a different score. We will not cover the methods here, but it is possible to efficiently find a maximum among these many possibilities. By efficiently, we mean some algorithmic method that avoids computing each possibility in full detail, and then compares the entire collection to find the maximum. For example, to find the maximum among possible permutations, one can make use of the "Weighted Assignment Problem" solution, which provides an $O\left(n^3\right)$ solution to the problem of finding a match among n pairs of features. To find the optimum among all possible normalized models, one can make use of the geometric hashing approach to object recognition. However, these are issues of efficiency, and our focus here is on the formulation.

We have seen, in this formulation, the importance of the predicted variability in features in the models, since the distribution functions essentially provide the metric that is used to score potential matches. Although we have used the term variability to denote the probability density functions and the probability of non-obscuration, others might term these probabilities as uncertainties.

5 Some results

So that this paper is not completely devoid of experimental results, we show an example of object recognition that makes use of the foregoing formulation of match metrics. These results are from the thesis of Jyh-Jong Liu [5,6]. Figure 8 shows a portion of an infrared image of a number of vehicles, with extracted features overlaid

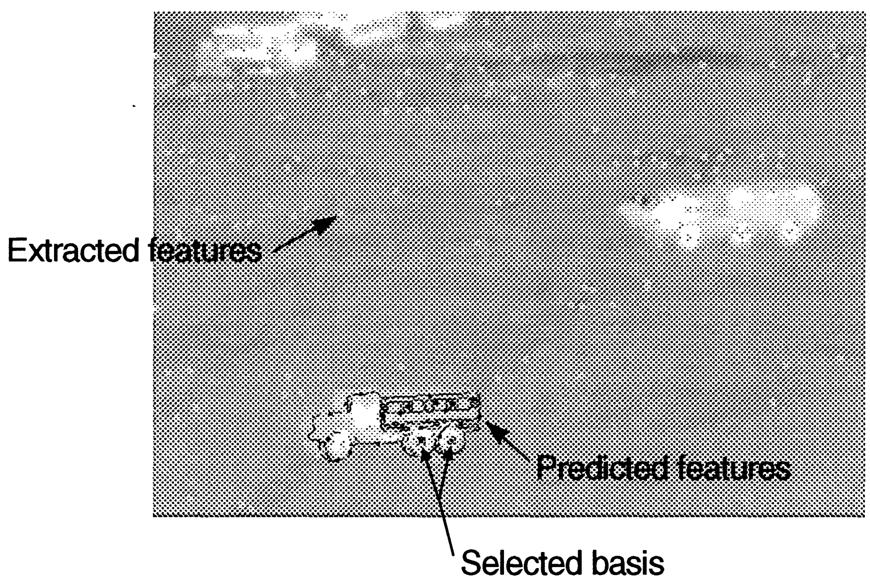

Figure 8. Recognition of an M35.

on the original image. The features were obtained as midpoints of line segments that have been fitted to the output of an edge extractor applied to the image. Each feature contains position information as well as an orientation value. In addition, a circle detector has been applied to the original image, and a collection of circle features have been extracted, as can be seen at the centers of the wheels. The circle features are represented by their centers and a radius value.

A database of over 80 models were processed in a similar fashion, where the models came from synthetic imagery produced to simulate views of vehicles from different aspects. CAD-CAM models of some of the vehicles were used to produce the synthetic images. Note, however, that a separate model is needed for each possible aspect, and so among the 80 models, there are really only about 10 different vehicle types.

Using basis sets formed by pairs of circle centers, all 80 models were processed to form a database of hundreds of normalized models. In the scene of extracted features, pairs of extracted circle centers were selected as a basis set, and used to normalize the extracted set of features. The resulting normalized scene to normalized model problem provides a 2-D translation and 2-D rotation invariant recognition problem.

We see the model with the highest vote overlaid on the original image, showing recognition of the M35 truck, among the dozens of possible models.

6 Extracted variability can represent uncertainty

We have discussed in great detail the importance of the predicted variability in the predicted features. What about extracted variability? So far, we have assumed that the extracted pattern consists of a collection of extracted, precise, features: $\{\mathbf{x}_i\}_{i=1}^S$. Suppose that instead, each extracted feature is a random variable with an associated pdf, say $\{(\mathbf{x}_i, g_i)\}_{i=1}^S$, where each extracted feature is represented by a distribution $g_i(\cdot - \mathbf{x}_i)$. Once again, these distributions will often be represented by local Gaussian functions. Also note that this "uncertainty" in the extraction is not caused by model variations, but rather uncertainty due to the sensor and feature extraction process. Further, each extracted feature can be associated with a probability that the feature is present at all, although we will not pursue this option here.

However, using the pdf for the extracted feature, we can consider two possibilities for incorporating this uncertainty into the object recognition problem. Each method has a considerably different goal.

In the first method, we reconsider the term $\log(\Pr(E_i|H)/\Pr(E_i))$ in light of the fact that the evidence is now the pair of pdf's $((\beta_i, \mathbf{y}_i, f_i), (\mathbf{x}_i, g_i))$. The main influence is in the numerator, where the evaluation of the probability of the evidence, given the match, becomes a convolution:

$$\Pr(E_i|H) = \int f_i(\mathbf{x} - \mathbf{y}_i) \cdot g_i(\mathbf{x} - \mathbf{x}_i) d\mathbf{x} = f_i * \breve{g}_i(\mathbf{x}_i - \mathbf{y}_i).$$

Here, $\breve{g}_i(x) = g_i(-x)$. Thus, rather than evaluating the predicted pdf at the extracted feature location, we convolve the predicted pdf with the extraction pdf, and evaluate at the offset of the predicted to extracted features. Note that if f and g are Gaussian, then the result of the convolution is a Gaussian whose covariance is the sum of the component covariances. What this says is that the predicted variation should be added to the extracted variation, in the sense of covariances, in order to assess the effective variability in order to compute the degree of match between the pair of features. Thus when we said before that the predicted variability should not include the sensor uncertainties and extraction variability, we can now amend this idea to say that the predicted variability can account for the extraction uncertainty, providing the extracted features are then point locations giving the nominal location of the feature.

For the denominator, the background density evaluation is basically the same:

$$\Pr(E_i) = \int f_i(\mathbf{x} - \mathbf{x}_i) \cdot \rho(\mathbf{x}) d\mathbf{x} .$$

These formulas provide a revised formula for the computation of the match metric Q, in a straightforward fashion.

The other method for dealing with extraction uncertainty views the previous match metric score Q to be a function of the extraction data: $Q = Q\left(\left\{\mathbf{x}_i\right\}_{i=1}^{S}\right)$. Then, as we vary the extracted features $\mathbf{x}_i$ with sampling density $g_i(\cdot - \mathbf{x}_i)$, we can then trace forward to view Q as a random variable (Figure 9), whose statistics can be

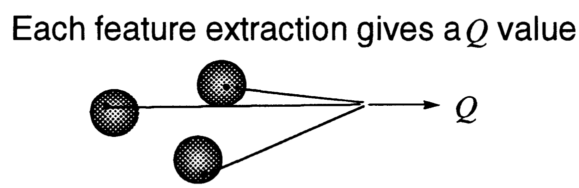

Figure 9. Q as a random variable.

approximated analytically, or computed by a Monte Carlo simulation. Note that the analytic solution might be complicated by the fact that the best set of associations of extracted features to predicted features might vary as the $\mathbf{x}_i$ vary. If the variations are small relative to the distances between the predicted features, then this potential re-ordering of the features can be ignored.

Accordingly, in this method, we can view the output of the evaluation of any given hypothesis H not as a match metric score Q, but rather as a random variable Q with an associated pdf h. This modification will complicate the problem of maximum likelihood determination, since it is no longer so simple to find the "best" hypothesis. Furthermore, the geometric hashing methods and other efficient algorithmic implementations may become more complicated. However, the largest challenge is to determine how to use the pdf of the score Q.

The concept of using a probability-valued random variable to represent degree of uncertainty in a probability appears in other contexts. The important point to realize is that the sample space over which the random variable is defined is a different space than the one that is used to define the probability. The Q value relates to the posteriori probability of an hypothesis, and is an estimate related to a frequency of occurrence of a particular object in a particular location conditioned on the current set of data. The sample space defining the Q random variable is, in this case, the tuple of random variables representing the extracted features, although we could imagine other sample spaces that could also result in defining Q as a random variable as opposed to a simple variable.

To illustrate how the random variable nature of Q might be used, consider two random variables, $Q(H_1)$ and $Q(H_2)$ corresponding to two competing hypotheses. If the two Q's were simple variables, then we would make the decision that the correct hypothesis is the larger of the two. Since the Q's are random variables with associated pdf's: $f_{Q(H_1)}(\cdot)$ and $f_{Q(H_2)}(\cdot)$, we might instead decide on H_1 or H_2 according to whether

$$\iint_{s_1>s_2} f_{Q(H_1)}(s_1)\cdot f_{Q(H_2)}(s_2)ds_1ds_2 \underset{>}{<} \iint_{s_2>s_1} f_{Q(H_1)}(s_1)\cdot f_{Q(H_2)}(s_2)ds_1ds_2 .$$

Similar decisions rules can be written when there are three or more competing hypotheses. Other decision methods might make use of the degree of separation between the s-values in these integrals, and the relative costs of making one decision against another decision.

7 Other measures of uncertainty

We have seen how to obtain a measure of the quality of an hypothesis $Q(H)$, using the variability in the predicted features, and potentially using the variability in the

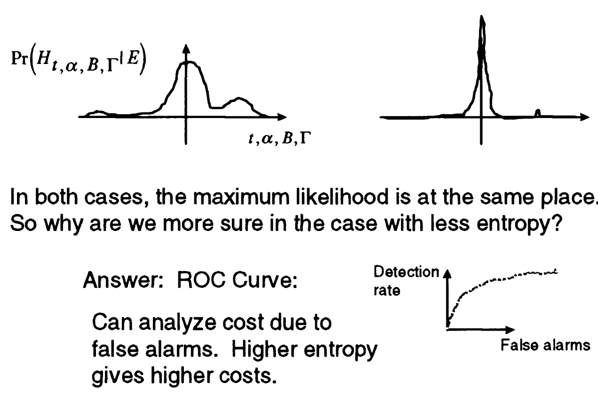

Figure 10. Entropy as a measure of uncertainty.

extracted features. Normally, we view the Q-value as a variable, although we have also seen how Q might become a random variable. In any case, maximum likelihood object recognition usually requires that we maximize $Q(H)$ over all possible hypotheses. There may be many possible hypotheses, which can include the position and orientation of the posited object, articulation parameters, and other model parameters, as well as the permutation of associations of model features to observed features. In general, H can be a combination of discrete and continuous variables; let us index H by the index t of the model type, α to specify articulation parameters, B to specify the pose parameters of the model, and Γ to specify the association of predicted features to extracted features. Then if we plot $Q(H_{t,\alpha,B,\Gamma})$ (or the functionally-dependent probability value) as a function of the index, we can consider the output of the object recognition system to be the index where the graph is maximized. However, we can additionally ask about the "peakedness" of the maximum.

Clearly, if the entropy is large, and the peak narrow, we should be more confident of the maximum index (see Figure 10). A narrow peak provides confidence in the actual values of the indices that are declared as the winning recognition parameters. For example, a measure of the entropy of the distribution, or a half-width measure of the peak, might be used as a measure of the "certainty" in the result of the object recognition system. This, of course, is a much different notion of certainty than we have discussed heretofore. On the other hand, one might be interested in all parameters that contribute to a aggregate hypothesis H that leads to a uniform action, and in this interpretation, we might not care if the peak is broad.

The question becomes, how should we evaluate the probability value as a function of the hypothesis indices? If we could compute the quality function over all possible indices, and then convert to a true probability distribution (which involves something akin to the partition function in physics), then we might use a problem-dependent model of the costs of certain decisions to compute an optimal strategy. Usually, the cost analysis makes use of the computation of the probability of detection of an object versus the probability of declaring a false alarm. The operating condition of a system (i.e., probability of detection versus the false alarm rate) can be computed using the probability function, and a prior distribution of hypotheses.

We won't carry out such an analysis here, especially because the conversion from a function $Q(H_{t,\alpha,B,\Gamma})$ to a function $\Pr(H_{t,\alpha,B,\Gamma} \mid E)$ might well be problematic. However, our main point is that the peakedness of the Q-function relates to the certainty of the recognition, and that costs of making the wrong declaration of model parameters relate to the evaluation of the degree of certainty.

To conclude, we see that the modeling and extraction uncertainty can be handled in an image processing object recognition system in reasonably logical ways. However, the degree of output certainty in a recognition declaration is not well-formulated by a the Bayesian analysis that we have used here, and further research and experience will be necessary to better evaluate methods of specifying and exploiting certainty in the results of such a system.

References

[1] Hummel, R., Landy, M., A statistical viewpoint on the theory of evidence, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1988, pp. 235-247.

[2] Hummel, R., Manevitz, L., Combination Calculi for Uncertainty Reasoning: Representing Uncertainty Using Distributions, To appear in *Annals of AI and Mathematics*, 1996, available by anonymous ftp to cs.nyu.edu under pub/local/hummel/papers/uncertainty.ps.Z.

[3] Ayache, N. and O. Faugeras, HYPER: A New Approach for the Recognition and Positioning of Two-dimensional Objects, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**(1), 1986, pp. 44-54.

[4] Rigoutsos, I., R. Hummel, Massively parallel model matching: Geometric hashing on the Connection Machine, *IEEE Computer* **25**(2), February, 1992, pp. 33-42.

[5] Jyh-Jong Liu, A model-based 3-D object recognition system using geometric hashing with attributed features, Ph.D. thesis, October, 1995, available by anonymous ftp to cs.nyu.edu, under pub/theses/liu.tar.Z.

[6] Rigoutsos and Hummel, A Bayesian approach to model matching with geometric hashing, *Computer vision and image processing*, Vol. **62**(1), 1995, pp. 11-26.