# *RECOGNITION - MATCHING*

# On the Verification of Hypothesized Matches in Model-Based Recognition[1]

W. Eric L. Grimson
MIT Artificial Intelligence Laboratory
545 Technology Square, Cambridge, Mass. 02139

Daniel P. Huttenlocher
Department of Computer Science
Cornell University, Ithaca, NY 14853

**Abstract**

Model-based recognition methods often use *ad hoc* techniques to decide if a match of data to a model is correct. Generally an empirically determined threshold is placed on the fraction of model features that must be matched. We instead rigorously derive conditions under which to accept a match. We obtain an expression relating the probability of a match occurring at random to the fraction of model features accounted for by the match, as a function of the number of model features, the number of image features, and a bound on the degree of sensor noise.

Our analysis implies that a proper matching threshold must vary with the number of model and data features, and thus should be set as a function of a particular matching problem rather than using a predetermined value. We analyze some existing recognition systems and find that our method predicts thresholds similar those determined empirically, supporting the technique's validity.

## 1. Introduction

A central problem in machine vision is recognizing partially occluded objects from noisy data. Recognition systems generally search for a match between elements of an object model and instances of those elements in the data, thereby recovering a transformation that maps part of the model onto part of the image. Approaches to model-based recognition (see [3, 2] for reviews) include clustering in parameter space (e.g. [17, 18]), searching a tree of corresponding model and image features (e.g., [9, 13, 5, 16, 1, 6, 15]), and directly searching for possible model-to-image transformations (e.g.,[8, 14]). These approaches all must decide if an object is present or absent on the basis of geometric evidence acquired from the sensory input. Here, we analyze this decision process and develop a formal means for deciding when a match should be accepted as correct.

Most recognition systems use *ad hoc* methods to determine what constitutes an acceptable match of a model to an image. For example, many systems order the possible interpretations of the data in terms of some measure of completeness, (e.g. the percentage of the model accounted for), and accept the best interpretations under this measure. If instances of the object model are present in the scene, this approach generally will find them. If no instance of the object is present, the interpretations that best account for the data are in fact incorrect. In this case, one must either accept false interpretations or have some means of deciding if the object is present. To reduce the computational complexity of recognition, the measure of completeness is often used to terminate the search once an interpretation is found that exceeds some empirically determined threshold. Once again, it is necessary to determine if an interpretation is good enough to accept as correct.

[1]Research funded in part by ONR URI grant N00014-86-K-0685, in part by NSF Grant IRI-8900267, and in part by DARPA under Army contract DACA76-85-C-0010 and ONR contract N00014-85-K-0124.

Current methods for deciding if a match is correct are based on empirically determined thresholds. In this paper we instead rigorously analyze what constitutes a good match of a model to an image. Specifically, we address the following question:

> Suppose that we are given a model with $m$ features, a set of $s$ data features, and bounds $\epsilon_p$ and $\epsilon_a$ on the positional and orientational error in the data. Further, suppose that some recognition method has found a match accounting for a fraction $f$ ($f \in [0,1]$) of the $m$ model features. What is the relation between $f$ and the likelihood $\delta$ that such a match can occur at random?

We use this relation to set a threshold on the minimum fraction of model features that must be matched, $f_0$, so that the likelihood of such a match occurring at random is small (e.g., $\delta < .001$). Note that there may not be a value of $f_0$ for all choices of $\delta$ (e.g., as $\delta$ gets very small, or as $m$, $s$, $\epsilon_p$, or $\epsilon_a$ get very large there may be no fraction of model features that limits the probability of a random match to $\delta$).

There are three basic steps to our analysis. First, given a particular feature type, the type of transformation from model to image, and a bound on the sensor error, we characterize the set of transformations consistent with a single pairing of a model and image feature. This set of transformations defines a volume $V$ in the *transformation space*, $\mathcal{P}$ (a $d$-dimensional space with one dimension for each of the $d$ parameters of the transformation). We then use an occupancy model [7] to determine the probability, $Pr\{v \geq l\}$ that the number of volumes intersecting at a common point in transformation space is at least $l$. This provides an estimate of how often a match of $l$ features will occur at random. Finally, the probability that $l$ volumes will intersect at random is used to set a threshold on the minimum fraction of model features, $f_0$, that must be matched in order for an interpretation to have a small probability of occuring at random.

## 2. The Space of Transformations

A rigid object's pose is characterized by a transformation from model to sensor coordinates. We focus on the case of a similarity transformation (i.e., a translation, rotation, and scaling). The set of possible poses can be viewed as a *transformation space* having one dimension for each parameter of the transformation from model to sensor coordinates. A point in this transformation space defines a pose of an object, which in turn defines a possible solution to the recognition problem. For example, with a 2D image and world, the transformation space is 4D (translation in $x$ and $y$, planar rotation, and scaling).

A match of a model feature and an image feature (e.g., an edge or vertex) defines a range of possible transformations, i.e., a volume in the transformation space. The size and shape of this volume depends on the type and accuracy of the feature. In this section we present an analytic expression for the size of this volume. The development is similar to that in [11], however here we consider a continuous space as opposed to one that is uniformly tesselated. The discussion is limited to the case of 2D problems where the transformation is an isometry (translation and rotation without scaling), and the features are linear edge fragments or points. A similar analysis holds for 3D problems and for problems involving change of scale, and is described in [12].

Consider the problem of recognizing a two-dimensional polygonal model from noisy, occluded data. We let $\mathbf{M}_J$ be the vector from the origin to the midpoint of the $J^{th}$ model edge, $\hat{\mathbf{T}}_J$ be the unit tangent of the edge, and $L_J$ be the length of the edge, all measured in the model coordinate system, $\mathcal{M}$. We let $\mathbf{m}_j, \hat{\mathbf{t}}_j, \ell_j$ denote similar parameters for the

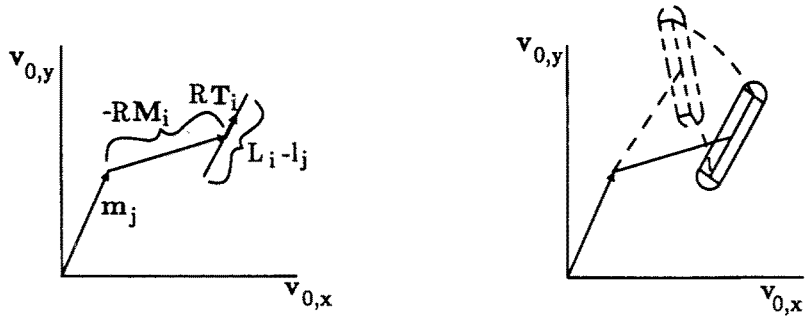


Figure 1: The range of feasible translations. Left: fixed $\theta$ without error, the line in direction $R\mathbf{T}_i$ denotes set of feasible translations. Right: allowing error, the region enclosed in solid lines shows slice $\mathcal{S}(\theta, j, J)$ for a particular $\theta$. Dashed region shows helical path of slice as $\theta$ varies.

---

$j^{th}$ data edge, measured in the sensor coordinate system, $\mathcal{I}$. (Note that we use upper case for model parameters and lower case for data parameters.)

The transformation from model to sensor coordinates may be represented by

$$\mathbf{v}_s = R_\theta \mathbf{V}_M + \mathbf{V}_0$$

where $\mathbf{V}_M$ is a vector in model coordinates, $R_\theta$ is a rotation matrix of angle $\theta$, $\mathbf{V}_0$ is a translation offset, and $\mathbf{v}_s$ is the corresponding vector in sensor coordinates.

What transformations will map a model edge to a data edge? If $\ell_j > L_J$, the two edges cannot match. Thus, suppose that $\ell_j \leq L_J$. Then the rotation matrix $R_{\theta_m}$ is defined by the angle $\theta_m$ between $\hat{\mathbf{T}}_J$ and $\hat{\mathbf{t}}_j$. Many translations will cause the edges to overlap, because $\ell_j \leq L_J$. If one endpoint of the data edge coincides with a transformed model edge endpoint, the translation is the difference between them:

$$\mathbf{V}_0 = \mathbf{m}_j - R_{\theta_m}\mathbf{M}_J \pm \frac{L_J - \ell_j}{2} R_{\theta_m}\hat{\mathbf{T}}_J$$

where the $\pm$ indicates the two cases. Because any intermediate position is also acceptable, the set of translations consistent with matching model edge $J$ to data edge $j$ is

$$\left\{ \mathbf{m}_j - R_{\theta_m}\mathbf{M}_J + \gamma R_{\theta_m}\hat{\mathbf{T}}_J \;\middle|\; \gamma \in \left[ -\frac{L_J - \ell_j}{2}, \frac{L_J - \ell_j}{2} \right] \right\}. \tag{1}$$

Hence, matching model edge $J$ to data edge $j$ gives a set of points in transformation space, with a single value for the rotation and a set of values for the translation, corresponding to a line of length $L_J - \ell_j$, with orientation $R_{\theta_m}\hat{\mathbf{T}}_J$ in the $x$–$y$ plane (Figure 1 Left).

This ignores the issue of noise in the measurements. In practice, we may only know the position of the data edge's endpoints to within some ball of radius $\epsilon_p$, and the orientation to within an angular error of $\epsilon_a$. For 2D lines, these error ranges are related. Given endpoint variations of $\epsilon_p$, the maximum angular variation occurs when the correct line is tangent to circles of radius $\epsilon_p$ about the two endpoints, and provided $\ell > 2\epsilon_p$, is given by

$$\epsilon_a = \tan^{-1}\left( \frac{2\epsilon_p}{\sqrt{\ell^2 - 4\epsilon_p^2}} \right).$$

Inclusion of error effects on position measurements imply that the line of feasible translations, for a given rotation, (equation (1)), must be expanded to include any points

in the parameter space within $\epsilon_p$ of that line. Further, this expansion into a region must be repeated for each value of $\theta$ in $[\theta_m - \epsilon_a, \theta_m + \epsilon_a]$. Note that this carves out a skewed volume in transformation space [4], because the region's center and orientation are functions of $\theta$, as illustrated in Figure 1 Right.

Thus given $\mathbf{M}_J, \hat{\mathbf{T}}_J, L_J, \mathbf{m}_j, \hat{\mathbf{t}}_j$, and $\ell_j$, if $\ell_j - 2\epsilon_p > L_J$ then there are no consistent transformations, otherwise the set of feasible transformations is denoted by the volume

$$\mathcal{V}(j,J) = \bigcup_{\theta \in [\theta_m - \epsilon_a, \theta_m + \epsilon_a]} \mathcal{S}(\theta, j, J)$$

where an individual set of translations is denoted by:

$$\mathcal{S}(\theta,j,J) = \left\{ (\theta, \mathbf{V}_0) \,\middle|\, \exists \gamma, |\gamma| \leq \frac{L_J - \ell_j}{2}, \|\mathbf{m}_j - R_\theta \mathbf{M}_J + \gamma R_\theta \hat{\mathbf{T}}_J - \mathbf{V}_0\| \leq \epsilon_p \right\}.$$

Since each slice $\mathcal{S}(\theta, j, J)$ consists of two hemicircles and a rectangle, it is easy to show that the volume of the region $\mathcal{V}(j, J)$ is given by

$$c_{jJ} = 2\epsilon_a \left[ 2\epsilon_p (L_J - \ell_j) + \pi \epsilon_p^2 \right].$$

The term in braces is the area of one slice. Integration over a range of angles yields the $2\epsilon_a$ term. For simplicity, we let the data edge have a length $\ell_j = (1 - \alpha_{jJ}) L_J$ where $\alpha_{jJ}$ denotes the amount of occlusion of the edge, so that the volume is

$$c_{jJ} = 2\epsilon_a \left[ 2\epsilon_p \alpha_{jJ} L_J + \pi \epsilon_p^2 \right]. \tag{2}$$

If we are dealing with point features, rather than extended edges, the above result can be specialized. Here $L_j \to 0$ so that equation (2) becomes $c_{jJ} = 2\epsilon_a \pi \epsilon_p^2$. What is $\epsilon_a$ in the case of a point feature? For a vertex, its orientation is the direction of the bisector of the two edges defining the vertex, and hence $\epsilon_a$ is a bound on the error in measuring that orientation. For a curvature extremum or inflection, the local tangent of the curve defines the orientation, and $\epsilon_a$ is again defined by a bound on measuring this orientation. For truly isolated points, $\epsilon_a = \pi$. In any event, our analysis provides estimates for $c_{jJ}$ both for edge features and for vertices.

For the case of a rigid 2D isometric transformation, we have characterized in (2) the volume of transformation space, $c_{jJ}$, consistent with a single data-model pairing $(j, J)$. The expression is a function of the noise in the data measurements, $\epsilon_p$ and $\epsilon_a$, and in the case of edges is further a function of the amount of occlusion, $\alpha_{jJ}$, and the length of the model edge, $L_J$. In [12] we consider adding scaling to the transformation as well as the case of 3D transformations. We now turn to the question of how these volumes interact.

## 3. The Probability of a Conspiracy

The intersection, if any, of two volumes in transformation space defines the set of transformations consistent with both pairings. Thus a correct match of a model to an image will lie in the intersection of several volumes. In this section we consider the likelihood that $l$ volumes in transformation space will intersect at random. Such an event corresponds to an arrangement of image features that happens to be consistent, within error, with $l$ of the model features, but which does not actually correspond to an instance of the object.

The likelihood that $l$ transformation space volumes will intersect at random is a function of their number and size. The number depends on the number of model and image features. The size depends on the noise, the feature type, and for edge features, the

amount of occlusion. To be confident that a match with $l$ model features is correct, we would like $l$ to be large enough that a random matching of that size is very unlikely.

To characterize the likelihood that several volumes will intersect at random we use a statistical occupancy model. In the discrete case, if $r$ events are uniformly randomly distributed across $n$ buckets, an occupancy model can be used to estimate the probability that a given bucket will contain $k$ events. The events in our case are points in the volumes in transformation space, and the buckets are points in the transformation space itself. These quantities are continuous, and thus we consider the limiting case as $n, r \to \infty$.

The volume of transformation space defined by each incorrect model and image feature pairing is independent of the correct match. We assume that the image features are also independent of one another, so we can model the volumes in transformation space as independent random events. The distribution of these volumes depends on the image features, which are unknown, so we assume the uniform distribution as an approximation.

While the volumes in transformation space can reasonably be viewed as independent random events, we are modeling the probability of events occurring at points in these volumes. As the number of volumes, $R$, gets large (compared with the ratio of the total size of the transformation space to the size of each volume, $V/c$) the overall distribution of points in the space also is random. For the cases of interest here $Rc \gg V$, so the assumption of independent random pointwise events is a reasonable approximation.

Given a uniform random distribution of $r$ events into $n$ cells, a number of different statistical occupancy models can be used to characterize the likelihood, $p_k$, that a given cell will contain exactly $k$ events. We use the Bose-Einstein statistic, where it is assumed that each assignment of counts to cells occurs with equal probability [7]. Under this model, for large $r$ and $n$, where $\frac{r}{n} \to \lambda$, the limiting case is the geometric distribution,

$$p_k \approx \frac{\lambda^k}{(1+\lambda)^{k+1}} \approx \frac{1}{1+\lambda}\left(\frac{1}{1+\frac{1}{\lambda}}\right)^k. \tag{3}$$

We are interested in establishing conservative bounds on the likelihood that a large number of volumes will intersect at random, thus we use the Bose-Einstein statistic because it provides a higher estimate of this likelihood than do other models.

The parameter $\lambda$ of the occupancy model is the ratio of the occupied volumes of the transformation space to the total size of the transformation space. From equation (2) we know that each pair of model and image features defines a volume of size $c_{jJ}$ in transformation space. There are $ms$ such volumes for $m$ model features and $s$ image features, so the occupied volume of the transformation space is given by the sum of $c_{jJ}$ for $j = 1, ..., s$ and $J = 1, ..., m$.

The total size of the transformation space is just the product of the ranges for the space's dimensions. Each rotational dimension ranges over $[0, 2\pi]$, and each translational dimension ranges over $[0, D]$, where $D$ is the linear extent of the image. Thus for a 2D isometry (translation and rotation) we get

$$\lambda = \frac{\sum_{j=1}^{s}\sum_{J=1}^{m} c_{jJ}}{2\pi D^2} = ms\overline{c},$$

where $\overline{c}$ is the average normalized volume size. For 2D edges, from equation (2) we obtain

$$\overline{c} = \frac{2\epsilon_a\left[2\epsilon_p\overline{\alpha}\overline{L} + \pi\epsilon_p^2\right]}{2\pi D^2} = \frac{\epsilon_a}{\pi}\left[2\overline{\alpha}\frac{\epsilon_p}{D}\frac{\overline{L}}{D} + \pi\left(\frac{\epsilon_p}{D}\right)^2\right] \tag{4}$$

where $\overline{L}$ is the average edge length and $\overline{\alpha}$ is the average amount of occlusion of the edges (the average value of $\alpha_{jJ}$). As expected $\overline{c}$ increases as the noise $\epsilon_a, \epsilon_p$ increases, and as

the average amount of occlusion of the edges $\overline{\alpha}$ increases. In the case of 2D points (with associated orientations), the average normalized volume size is $\overline{c} = \epsilon_a \epsilon_p^2 / D^2$. Note that we can restrict $\epsilon_a \leq \pi$ and $\epsilon_p \leq \frac{D}{2}$. In the extreme, this can lead to $\overline{c} > 1$, which does not make physical sense. We should really take the minimum of the above expressions and unity, but in practice $\overline{c} \ll 1$ and hence we ignore this special case.

A particular recognition task thus defines a value for $\lambda$, based on the type of transformation from model to image, the type of features, the number of model features, $m$, and data features, $s$, and a bound on the positional and angular error, $\epsilon_p$ and $\epsilon_a$. Given a value for $\lambda$, the probability that $l$ or more of the volumes intersect at random is given by

$$Pr\{v \geq l\} = 1 - \sum_{k=0}^{l-1} p_k. \tag{5}$$

This corresponds to an arrangement of data features occurring at random such that $l$ pairs of model and data features are consistent with one another (within the error bounds). From $Pr\{v \geq l\}$ we can determine the fraction of model features, $f_0$, such that the probability of $mf_0$ features being matched at random is less than some predefined level, $\delta$. This value is just the smallest $f$ such that $Pr\{v \geq mf\} \leq \delta$, i.e.,

$$f_0 = \min\{f \mid Pr\{v \geq mf\} \leq \delta\}. \tag{6}$$

## 4. Deriving Formal Thresholds

We have used an occupancy model to determine an expression for the probability that $l$ or more volumes in transformation space will intersect at random, as a function of the number of features, the type of features, and bounds on the sensor error. The expression was then used to set a threshold, $f_0$, on the fraction of model features that must be matched in order to limit the probability of a random matching to some level. In this section we derive a closed-form expression for $f_0$.

The probability that there will be $l$ or more events occurring at random at a point in transformation space is given by (5). Thus to distinguish a correct interpretation from a random one we set the threshold, $f_0$, such that the probability of $l = mf_0$ events coinciding at random is less than $\delta$. Substituting $mf_0$ for $l$ and equation (3) for $p_k$ in (5), we obtain

$$Pr\{v \geq mf\} = 1 - \sum_{k=0}^{mf_0-1} \frac{\lambda^k}{(1+\lambda)^{k+1}} \leq \delta.$$

Using the geometric series relationship, we can isolate $f_0$ by appropriate algebra:

$$f_0 \geq \frac{\log\left(\frac{1}{\delta}\right)}{m \log\left(1 + \frac{1}{ms\overline{c}}\right)}. \tag{7}$$

Thus to obtain a value for the fraction of model features that must be matched in order to limit the probability of a random conspiracy to $\delta$, we simply need to compute $\overline{c}$ for the particular parameters of our recognition task, and then use (7) to compute $f_0$. The value of $\overline{c}$ depends on the particular type of feature being matched and the bounds on the sensor error. In the case of 2D edge fragments, we derived $\overline{c}$ in (4).

Note that equation (7) exhibits the expected behavior. If the noise in the data increases, then $\overline{c}$ increases, and so does the bound on $f_0$. Similarly, as the amount of occlusion increases, then so does $\overline{c}$ and thus the bound on $f_0$. As either $m$ or $s$ increases so does the bound on $f_0$, and as $\delta$ decreases $f_0$ increases. Also note that for large values of $ms$, one gets the approximation $f_0 \geq s\overline{c} \log 1/\delta$. Thus, in the limit, the bound on the

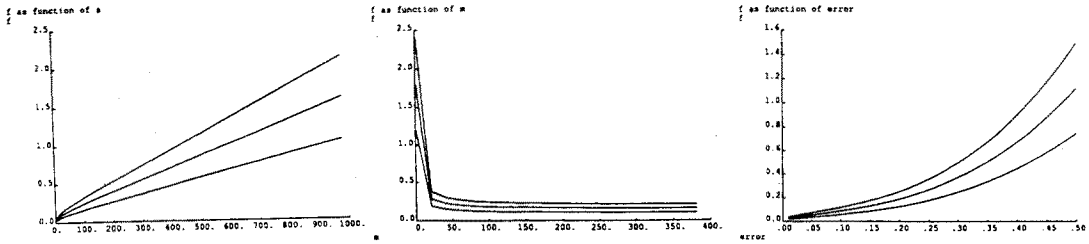

Figure 2: Bounds on threshold. Left: $f_0$ as a function of $s$. Middle: $f_0$ as a function of $m$. Right: $f_0$ as a function of error. Percentage of error along horizontal axis $p$ defines sensing errors of $\epsilon_a = p\pi$ and $\epsilon_p = p\overline{L}$. The three plots in each case are for $\delta = .0001, .001, .01$ from top to bottom respectively.

---

fraction of the model is linear in the number of sensory features, linear in the average size of the volumes in transformation space, and varies logarithmically with the inverse probability of a false match.

The expression for $f_0$ in (7) can yield values that are greater than 1.0, which makes no sense as a *fraction* of the model features. When $f_0$ is greater than 1.0 it means that for the given number and type of features, and the given bounds on sensor error, it is not possible to limit the probability of a false match to the chosen $\delta$ (even if all the model features are matched to some sensor feature).

There are several possible choices for $\delta$. One could simply set $\delta$ to be some small number, e.g., $\delta = .001$, so that a false positive is likely to arise no more than one time in a thousand. One could also set $\delta$ as a function of the scene complexity, e.g., some multiple of the inverse of the total number of data model pairings, (i.e., $\frac{\beta}{ms}$.) A third possibility is to set $\delta$ so that the likelihood of a false positive, integrated over the entire transformation space, is small (e.g., $< 1$). The idea is to determine the appropriate value of $\delta$ such that one expects no random matches to occur. If we let $\nu$ be a measure of the system's sensitivity in distinguishing transformations, then we could choose $\delta = \nu/2\pi D^2$. For example, we could set $\nu$ to be a function of the noise in the data measurements, given by the product in uncertainties: $(2\epsilon_a)(\pi\epsilon_p^2)$. In this case, we get

$$f_0 \geq \frac{\log\left(\frac{D^2}{\epsilon_a \epsilon_p^2}\right)}{m \log\left(1 + \frac{1}{ms\overline{c}}\right)}. \tag{8}$$

We graph examples of $f_0$ in Figure 2. Figure (2 Left) displays values of $f_0$ as a function of $s$, with $m = 32$, $c = .0002215$ (these numbers are taken from the RAF system analyzed in section 5). Each graph is for a different value of $\delta$. Note that as $s$ gets large, the graphs become linear, as expected. Figure (2 Middle) displays $f_0$ as a function of $m$ for different values of $\delta$. Here, $s = 100$, $c = .0002215$. Note that as expected, when $m$ becomes large, $f_0$ becomes a constant independent of $m$. Figure (2 Right) displays $f_0$ as a function of the sensor error, for different values of $\delta$. Here, $s = 100$, $m = 32$. The percentage of error along the horizontal axis, $p$, is used to define sensing errors of $\epsilon_a = p\pi$ and $\epsilon_p = p\overline{L}$. As expected, the threshold on $f_0$ increases with increasing error.

We can modify our preceding analysis to handle weighted matching methods as well. One common scheme is to use the size of each data feature as a weight. In the case of 2D edges, for example, a data-model pairing $(j, J)$ would carry a weight of $\ell_j$ (the length of the data edge), so that transformations consistent with pairings of long data edges to

| Occlusion | $f$, eqn (8) | $f$, ($\delta = .001$) | $f$, ($\delta = .0001$) | $f$, ($\overline{\ell} = \overline{L}$) | $f$, ($\overline{\ell} = .75\overline{L}$) | $f$, ($\overline{\ell} = .5\overline{L}$) |
|---|---|---|---|---|---|---|
| 0.0 | 0.225 | 0.173 | 0.230 | 0.119 | 0.091 | 0.062 |
| 0.2 | 0.263 | 0.202 | 0.270 | 0.153 | 0.116 | 0.079 |
| 0.4 | 0.301 | 0.231 | 0.308 | 0.188 | 0.142 | 0.097 |
| 0.6 | 0.337 | 0.259 | 0.346 | 0.222 | 0.168 | 0.114 |
| 0.8 | 0.374 | 0.287 | 0.383 | 0.257 | 0.194 | 0.131 |
| 1.0 | 0.409 | 0.315 | 0.420 | | | |

Table 1: Predicted bounds on termination threshold, as a function of amount of occlusion. First three cases are unweighted, second three use edge length as a weight.

model edges would be more highly valued than those involving short data edges. Working through similar algebra [12], where $\overline{\ell}$ is the average length of the data edges, leads to:

$$f_0 \geq \frac{\log\left(\frac{1}{\delta}\right)}{m\overline{L}\log\left(1+\frac{1}{ms\overline{\ell}\overline{c}}\right)}. \tag{9}$$

## 5. Some Real World Examples

To demonstrate the utility of our method, we analyze some working recognition systems that utilize a threshold on the fraction of model features needed for a match. The analysis predicts thresholds close to those determined experimentally, suggesting that the technique can be profitably used to analytically determine thresholds for model-based matching. Because our analysis shows that the proper threshold *varies* with the number of model and data features, it is important to be able to set the threshold as a function of a particular matching problem rather than setting it once based on experimentation.

We first consider the interpretation tree method [13, 5, 16] for recognizing sets of 2D parts. In this approach, a tree of possible matching model and image features is constructed. Each level of the tree corresponds to an image feature. At every node of the tree there is a branch corresponding to each of the model features, plus a special branch that accounts for model features that do not match the image. A path from the root to a leaf node maps each image feature onto some model feature or the special "no-match" symbol. The tree is searched by maintaining pairwise consistency among the nodes along a path. Consistency is checked using distance and angle relations between the model and image features specified bythe nodes. If a given node is inconsistent with any node along the path to the root then the subtree below that point is pruned from further consideration.

A consistent path from the root to a leaf that accounts for more than some fraction of the model features is accepted as a correct match. This threshold is chosen experimentally. In our analysis of thresholds for the interpretation tree method, we use the parameters for the examples presented in [13]. These values are substituted into equation (2), and then a threshold $f_0$ is computed using equations (7) and (8). In the experiments reported in [13], the following parameters hold: $m = 32$, $s = 100$, $\overline{L} = 23.959$, $\epsilon_p = 10$, $\epsilon_a = \frac{\pi}{10}$.

We have computed $\overline{c}$ as a function of the amount of occlusion $\overline{\alpha}$, and then determined the corresponding threshold $f_0$ on the fraction of model features. Note that an occlusion of 1 represents the limiting case in which only a point on the line is visible. The results are given in Table 1. The first column of the table shows the values of $f_0$ computed using equation (8), where $\delta = \epsilon_a\epsilon_p^2/D^2$. For comparison, the second and third columns of the table are computed using equation (7), with the probability of a random match, $\delta$, set to .001 and .0001, respectively.

| Occlusion | Object-1 $f$, ($\epsilon_a = \frac{\pi}{10}$) | Object-1 $f$, ($\epsilon_a = \frac{\pi}{15}$) | Object-2 $f$, ($\epsilon_a = \frac{\pi}{10}$) | Object-2 $f$, ($\epsilon_a = \frac{\pi}{15}$) |
|---|---|---|---|---|
| 0.0 | 0.224 | 0.185 | 0.206 | 0.168 |
| 0.2 | 0.261 | 0.212 | 0.243 | 0.195 |
| 0.4 | 0.297 | 0.238 | 0.280 | 0.221 |
| 0.6 | 0.333 | 0.264 | 0.316 | 0.247 |
| 0.8 | 0.368 | 0.289 | 0.353 | 0.273 |

Table 2: Predicted bounds on termination threshold, as a function of the amount of occlusion, for the HYPER system.

As expected, the bound on $f$ increases as the amount of occlusion increases. Note that for occlusions ranging from none to all (0 to 1), the bound on $f$ only varies over a range of 0.225 to 0.409. Empirically, running the RAF system on a variety of images of this type [13] using thresholds of $f = .4$ resulted in no observed false positives, while using thresholds of $f = .25$ would often result in a few false positives. Since the occlusion was roughly .5, this observation fits nicely with the predictions of Table 1, i.e., a threshold of .4 should yield no errors, while a threshold of .25 cannot guarantee such success.

If we use the lengths of the data features to weight the individual feature matchings then using equation (9) in place of (8) leads to the predictions shown in the second part of Table 1. Again, this agrees with empirical experience for the RAF system, in which weighted matching using thresholds of $f = .25$ almost always led to no false positives, while using thresholds of $f = .10$ would often result in a few false positives.

Second, we consider the HYPER system [1]. HYPER also uses geometric constraints to find matches of data to models. An initial match between a long data edge and a corresponding model edge is used to estimate the transformation from model to data coordinates. This estimate is then used to predict a range of possible positions for unmatched model features, and the image is searched over this range for potential matches. Each potential match is evaluated using position and orientation constraints, and the best match within error bounds is added to the current interpretation. The additional model-data match is used to refine the transformation estimate, and the process is iterated.

Although not all of the parameters needed for our analysis are given in the paper, we can estimate many of them from the illustrations in the article. Given several estimates for the measurement error, a range of values for the threshold $f$ are listed in Table 2. Object-1 and Object-2 refer to the object labels used in [1]. In these examples, we use errors of $\epsilon_a = \pi/10$ and $\pi/15$ radians, and $\epsilon_p = 3$ pixels.

In HYPER, a threshold of .25 is used to discard false positives, and no false positives are observed during a series of experiments with the system. For the two objects listed in Table 2, HYPER found interpretations of the data accounting for a fraction of .55 of the model for Object-1 and accounting for a fraction of .40 of the model for Object-2. Both these observations are in agreement with the thresholds predicted in Table 2, for different estimates of the data error.

Thus for two different recognition systems (RAF and HYPER), using both weighted and unweighted matching schemes, we see that the technique developed here yields matching thresholds similar to those determined experimentally by the systems' designers.

## 6. Conclusion

In order to determine what constitutes an acceptable match of a model to an image, most recognition systems use an empirically determined threshold on the fraction of model fea-

tures that must be matched. We have developed a technique for analytically determining the fraction of model features $f_0$ that must be matched in order to limit the probability of a random conspiracy of the data to some level $\delta$. This fraction $f_0$ is a function of the feature type, the number of model features $m$, the number of sensor features $s$, and bounds on the translation error $\epsilon_p$ and the angular error $\epsilon_a$ of the features.

Our analysis shows that the proper threshold varies with the number of model and data features. A threshold that is appropriate for relatively few data features is not appropriate when there are many data features. Thus it is important to set the threshold as a function of a particular matching problem, rather than setting a single threshold based on experimentation. Our technique provides a straightforward means of computing a matching threshold for the values of $m$ and $s$ found in a given recognition situation.

# References


[1] Ayache, N. & O.D. Faugeras, 1986, "HYPER: A new approach for the recognition and positioning of two-dimensional objects," *IEEE Trans. PAMI* **8**(1), pp. 44–54.

[2] Besl, P.J. & R.C. Jain, 1985, "Three-dimensional object recognition," *ACM Computing Surveys* **17**(1), pp. 75–154.

[3] Chin, R.T. & C.R. Dyer, 1986, "Model-based recognition in robot vision," *ACM Computing Surveys* **18**(1), pp. 67–108.

[4] Clemens, D.T., 1986, The recognition of two-dimensional modeled objects-in images, M. Sc. Thesis, Massachusetts Institute of Technology, Electrical Engineering and Computer Science.

[5] Ettinger, G.J., 1988, "Large Hierarchical Object Recognition Using Libraries of Parameterized Model Sub-parts," *IEEE Conf. on Comp. Vis. & Pattern Recog.*, pp. 32–41.

[6] Faugeras, O.D. & M. Hebert, 1986, "The representation, recognition and locating of 3-D objects," *Int. J. Robotics Research* **5**(3), pp. 27–52.

[7] Feller, W., 1968, *An Introduction to Probability Theory and Its Applications,* New York, Wiley.

[8] Fischler, M.A. & R.C. Bolles, 1981, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM* **24**, pp. 381–395.

[9] Grimson, W.E.L., 1989, "On the Recognition of Parameterized 2D Objects," *International Journal of Computer Vision* **2**(4), pp. 353–372.

[10] Grimson, W.E.L., 1989, "The Combinatorics of Object Recognition in Cluttered Environments Using Constrained Search," *Artificial Intelligence*, to appear.

[11] Grimson, W.E.L. & D.P. Huttenlocher, 1990, "On the Sensitivity of the Hough Transform for Object Recognition," *IEEE Trans. PAMI*, to appear.

[12] Grimson, W.E.L. & D.P. Huttenlocher, 1989, On the Verification of Hypothesized Matches in Model-Based Recognition, MIT AI Lab Memo 1110.

[13] Grimson, W.E.L. & T. Lozano-Pérez, 1987, "Localizing overlapping parts by searching the interpretation tree," *IEEE Trans. PAMI* **9**(4), pp. 469–482.

[14] Huttenlocher, D.P. & S. Ullman, 1990, "Recognizing solid objects by alignment with an image," *Intl. J. of Computer Vision*, to appear.

[15] Ikeuchi, K., 1987, "Generating an interpretation tree from a CAD model for 3d-object recognition in bin-picking tasks," *International Journal of Computer Vision* **1**(2), pp. 145–165.

[16] Murray, D.W. & D.B. Cook, 1988, "Using the orientation of fragmentary 3D edge segments for polyhedral object recognition," *Intern. Journ. Computer Vision* **2**(2), pp. 153–169.

[17] Stockman, G., 1987, "Object recognition and localization via pose clustering," *Comp. Vision, Graphics, Image Proc.* **40**, pp. 361–387.

[18] Thompson, D.W. & J.L. Mundy, 1987, "Three-dimensional model matching from an unconstrained viewpoint," *Proc. Intern. Conf. Robotics & Automation*, Raleigh, NC, pp. 208–220.