MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 1395 October, 1992

Limitations of Geometric Hashing in the Presence of Gaussian Noise

Karen B. Sarachik

Abstract.
This paper presents a detailed error analysis of geometric hashing in the domain of 2D object recogition. Earlier analysis has shown that these methods are likely to produce false positive hypotheses when one allows for uniform bounded sensor error and moderate amounts of extraneous clutter points. These false positives must be removed by a subsequent verification step. Later work has incorporated an explicit 2D Gaussian instead of a bounded error model to improve performance of the hashing method.

The contribution of this paper is to analytically derive the probability of false positives and negatives as a function of the number of model features, image features, and occlusion, under the assumption of 2D Gaussian noise and a particular method of evidence accumulation. A distinguishing feature of this work is that we make no assumptions about prior distributions on the model space, nor do we assume even the presence of the model. The results are presented in the form of ROC (receiver-operating characteristic) curves, from which several results can be extracted; firstly, they demonstrate that the 2D Gaussian error model performs better for high clutter levels and degrades more gracefully as compared to the uniform bounded error model for the same conditions. They also directly indicate the optimal performance that can be achieved for a given clutter and occlusion rate, and how to choose the thresholds to achieve the desired rates.

Lastly, we verify these ROC curves in the domain of simulated images.

Keywords: Object recognition, Error analysis, Geometric Hashing, Gaussian error models

Acknowledgments: This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Army contract number DACA76-85-C-0010 and under Office of Naval Research contract N00014-91-J-4038.

1 Introduction

Geometric hashing is a technique introduced in [LSW87], [HW88], to solve the problem of recognizing objects and their associated poses in cluttered scenes. The main idea behind the technique is that instead of checking every possible correspondence of image to model features to establish a model pose and then checking the image for supporting evidence, the recognition process is considerably sped up by splitting it into two stages. In the first stage, a database of all possible views of the model are precomputed and stored in a hash table. Recognition consists of using $2D$ image features to index into the hash table in order to vote for possible model poses.

However, under the assumption of uniform bounded sensor error, performance degrades rapidly with even a moderate amount of clutter [GHJ91]. Intuitively, the reason is that the error causes the point entries in the hash table to blur into regions, making the table denser and increasing the chances that a random image point (i.e., a point not arising from the model) will corroborate an incorrect hypothesis.

In this paper we analyze the effect of a more realistic noise model on these techniques. The question we address in the paper is, what kind of performance can we expect from the techniques as a function of the number of model features and clutter features (*i.e.,* signal to noise ratio)?

To answer the question, first we briefly present the original hashing algorithms, then we show how to modify them in the presence of sensor error. We model the error as a $2D$ Gaussian distributed vector, which is often a more realistic model than the uniform bounded error model used in the earlier analysis [GHJ91]. A voting function for accumulating evidence for hypotheses based on the error model is presented. (Similar approaches to extending geometric hashing have been explored in [CHS90], [RH91].) This is the background for main question, which is, how does one determine a reliable point at which to separate correct from incorrect hypotheses? This question is relevant in the noiseless case as well: assume there is a 25% occlusion rate, and we are searching for a model of size 20. Do we decide that a hypothesis is true after seeing 15 corroborating features, or 12, or 10? Clearly, the lower the acceptance threshold, the higher the probability of false positives, and the higher the acceptance threshold, the higher the probability that we will miss a correct hypothesis, *i.e.* of false negatives.

To find the optimal acceptance threshold for a fixed occlusion rate and a fixed number of model and clutter points, we use the given error model and voting scheme to derive expressions for the probability density functions of weights of positive and negative hypotheses. We then vary the acceptance threshold and find the probability of false positives and true positives for that threshold. The results are plotted as ROC curves, which indicate the optimal performance that can be achieved for the given level of occlusion, clutter, and number of model points.

2 Statement of the Geometric Hashing Algorithm

We begin by reviewing the original geometric hashing algorithm assuming exact measurements [LSW87], [HW88]. The algorithm consists of two stages, a model preprocessing stage and a recognition stage. For simplicity, we restrict attention to planar objects in arbitrary 3D pose. The model representation consists of a set of (x, y) points in what we will call *model space*, which is simply some fixed coordinate system. The points can be corners, points of high curvature, or points of inflection of the $2D$ model.

Assuming orthographic projection, we can represent the image location $[u_i, v_i, 1]^T$ of each model point $[x_i, y_i, 1]^T$ with a simple linear transformation:

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

where the upper left of the transformation matrix is a 2×2 non-singular matrix, and $[t_x, t_y]^T$ is the translation vector. This is because the projection onto the $z = 0$ plane of a rotated, scaled, and translated point $(x, y, 0, 1)$ simplifies to

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot S \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix}$$

$$= S \begin{bmatrix} r_{11} & r_{12} & 0 & t_x \\ r_{21} & r_{22} & 0 & t_y \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix}$$

where S is a positive scale factor. It is a well known fact that if a point has coordinates $\vec{X}$ with respect to a given basis, then a linear transformation on the entire space leaves the coordinates of the point unchanged with respect to the transformed coordinates of the basis. The coordinates of $\vec{X}$ with respect to the basis are called *affine* coordinates, and it is their invariance under linear operations which is utilized in geometric hashing.

In the preprocessing stage, the hash table is constructed as follows: Every ordered triple of model points is used as a basis, and the affine coordinates (α, β) of all other model points are computed with respect to each basis. Thus, if $\vec{m}_0, \vec{m}_1$ and $\vec{m}_2$ are basis points, then we represent any other feature point by

$$\vec{m}_i = \vec{m}_0 + \alpha_i(\vec{m}_1 - \vec{m}_0) + \beta_i(\vec{m}_2 - \vec{m}_0).$$

The basis (*i.e.,* the 3 model points) is entered into the hash table at each (α_i, β_i) location. Intuitively, the invariance of the affine coordinates of model points with respect to 3 of its own points as basis is being used to "precompute" all possible views of the model in an image. The actual algorithm is:

- for every ordered model triplet $B_k = (m_0, m_1, m_2)$,
  - for every other model point m_j
    - (i) find coordinates $m_j = (\alpha_j, \beta_j)$ with respect to basis B_k

(ii) enter basis B_k at location (α_j, β_j) in the hash table.

The running time for this stage is $O(m^4)$, where m=number of model points.

At recognition time, the image is processed to extract $2D$ feature points which are used to index into the table. The choice of features used must be determined by what points were used as model feature points, *i.e.,* if corners were used as model features, then one might take the intersection of all line segments to be the image feature points. Every image triple is then taken to be a basis, and the affine coordinates of all other image points is computed with respect to the basis to index into the hash table and "vote" for all bases found there. Intuitively we are searching for any three image points which come from the model, and using the hash table to verify hypothesized triples of image points as instances of model points. Such an image triple will yield a large number of votes for its corresponding model basis. In particular:

- for every unordered image triplet (i_0, i_1, i_2)
  - (a) for every other image point i_j
    - (i) find coordinates $i_j = (\alpha_j, \beta_j)$ with respect to basis (i_0, i_1, i_2)
    - (ii) Index into the hash table at location (α_j, β_j) and increment a histogram count for all bases found there.
  - (b) If the weight of the vote for any basis B_k is sufficiently high, stop and output the correspondence between triple (i_0, i_1, i_2) and basis B_k as a correct hypothesis.

In some versions of the algorithm, the hypothesis that is output subsequently undergoes a verification stage before being accepted as correct. Note that we need to order the points either at the preprocessing stage or at recognition time, but not both (or there would be a six-fold redundancy of correspondences). We choose to order the points at the preprocessing stage and enter every model point with respect to a single unordered basis set 6 times, once for every ordering of the basis set. This makes the table 6 times denser, but then at recognition time we need only to choose an unordered image triple and impose a single arbitrary ordering upon it. This way, when we use the remaining image points to index into the hash table, we vote for the ordering of the model basis set as well as model basis set itself. The termination condition for accepting a correspondence of bases (and hence a pose of the object) and the confidence of the result are exactly the issues we investigate in this paper.

3 Modifications to the Algorithms in the Presence of Error

We now assume sensor uncertainty, namely, that a model feature appears at its projected location, but displaced by an error vector drawn from some distribution. Without noise, a correct matching (*i.e.,* a correct pairing of 3 model basis points and 3 image basis points) yields a single (x, y) location for a projected fourth model point in the image and a single (α, β) location for the same point in the hash table. Under the assumption of circular uniform bounded error, [GHJ91] showed that a matching gives rise to a circular disk of possible image locations for any projected fourth model point, and that this circular disk in the image translates to an ellipsoidal range of affine coordinates in the hash table. Therefore, in practice, the bases should be stored (weighted by some function of the error distribution) at all possible affine locations for the fourth point. However, it is simpler to analyze the probability that a uniformly distributed random point will fall into a given circle, than to translate the uniform distribution into a distribution on affine coordinates, and to analyze the probability that the random point with affine coordinates drawn from this distribution will fall into a given ellipse. It is clear that the answer is the same, but that the first space is more manageable than the second. We will therefore choose to do the analysis using the simpler space, keeping in mind that the results found in this fashion are true of the analysis done in hash table space as well. One consequence of this is that the analysis will apply equally well to alignment and to geometric hashing.

In the modified algorithm, instead of incrementing a histogram count for every eligible basis by a full vote, we increment the basis count by a number between 0 and 1 according to some "goodness" criterion, which in our case is a function of the distance of the point from its expected location. Because of this, we must look at the density function of the accumulated values for correct and incorrect hypotheses, instead of the discrete probability of a particular vote. We will use the term "weight of a hypothesis" to denote this concept.

4 Overview of the Analysis

The main claim of the paper is supported by the argument whose steps are as follows:

(a) A $2D$ circular Gaussian distribution often a more accurate model for sensor error, as opposed to a model assuming bounded uniform distribution [Wel91]. While a bounded model leads to conservative estimates on performance, a Gaussian model may lead to more practical estimates.

(b) Using this Gaussian distribution, the following is true: given a correspondence between 3 image points and 3 model points (referred to as a *hypothesis* for the rest of the article), and assuming a fixed standard deviation σ_0 for the sensed error of the image points, the location of a fourth model point with affine coordinates (α, β) (with respect to the 3 image basis points) will also have a $2D$ circular normal distribution with standard deviation σ_e:

$$\sigma_e = \sigma_0((1-\alpha-\beta)^2 + \alpha^2 + \beta^2 + 1)^{1/2}$$

Note that the possible distance of a fourth model point from its predicted location is now unbounded. In our scheme we will pick a cutoff search distance of $2\sigma_e$ for possible matching image features, which will imply a probability of false negative identification of 13.5% for a single point.

(c) As in [GHJ91], we find the density of σ_e, in one case when the values of σ_e come from a model appearing in the image ($f_H(\sigma_e)$), and in the other case, on σ_e resulting from incorrect hypotheses ($f_{\overline{H}}(\sigma_e)$). The two different density functions are

$$f_H(\sigma_e) = \left[\frac{1}{b_1\sigma_e}\right]^2 \qquad f_{\overline{H}}(\sigma_e) = \left[\frac{1}{b_0\sigma_e}\right]^4$$

where $b_1 = 0.58$, $b_0 = 0.35$.

(d) Next, we modify the recognition algorithm so that it assigns *weights* to points found within the error disk, as opposed to a single 1/0 vote. We choose to use:

$$v = \frac{1}{2\pi\sigma_e^2}e^{-\frac{d^2}{2\sigma_e^2}}$$

where d =distance from the point's hypothesized to actual location. This is the value of the $2D$ Gaussian density function whose center is at the hypothesized location.

(e) Define random variables V_H = the weight that a model point's projection contributes to its supporting basis, and $V_{\overline{H}}$ = the weight that a random image point contributes to a given basis. To demonstrate what this means, in the simpler bounded uniform error case, the distribution of V_H is:

$$f(V_H = v) = \begin{cases} (1-c) & v = 1 \\ c & v = 0 \\ 0 & \text{otherwise} \end{cases}$$

i.e., the probability that a fourth model point will contribute a weight of 1 to a correct hypothesis is $1-c$, where c is the probability of occlusion. A more complicated expression holds for $V_{\overline{H}}$ [GHJ91].

In the Gaussian error scheme with a cutoff distance of $2\sigma_e$ these distributions are:

$$f(V_H = v) = \begin{cases} c + e^{-2}(1-c) & \text{v=0} \\ \frac{(1-c)2\pi}{b_1^4}(s_2 - \frac{1}{e\sqrt{2\pi v}}) & \ell_1 < v \le \ell_2 \\ \frac{(1-c)2\pi}{b_1^4}\frac{e-1}{e\sqrt{2\pi v}} & \ell_2 < v \le \ell_3 \\ \frac{(1-c)2\pi}{b_1^4}(\frac{1}{\sqrt{2\pi v}} - s_1) & \ell_3 < v \le \ell_4 \\ 0 & \text{otherwise} \end{cases}$$

$$f(V_{\overline{H}} = v) = \begin{cases} 1 - \frac{4\pi}{R^2 b_0^4}[\frac{1}{s_1} - \frac{1}{s_2}] & \text{v=0} \\ \frac{2\pi}{R^2 b_0^4 v}(e\sqrt{2\pi v} - \frac{1}{s_2}) & \ell_1 < v \le \ell_2 \\ \frac{2\pi}{R^2 b_0^4 v}(e-1)\sqrt{2\pi v} & \ell_2 < v \le \ell_3 \\ \frac{2\pi}{R^2 b_0^4 v}(\frac{1}{s_1} - \sqrt{2\pi v}) & \ell_3 < v \le \ell_4 \\ 0 & \text{otherwise} \end{cases}$$

where

$$\ell_1 = \frac{1}{2\pi {s_2}^2 e^2} \qquad \ell_2 = \frac{1}{2\pi {s_2}^2}$$

$$\ell_3 = \frac{1}{2\pi {s_1}^2 e^2} \qquad \ell_4 = \frac{1}{2\pi {s_1}^2}$$

and s_1, s_2 are the minimum and maximum allowable values for σ_e, respectively.

(f) The probability density function for the weight of an incorrect hypothesis is calculated as follows: For a single random point in an image with m projected model points, the distribution is:

$$f(V_{\overline{H}_m} = v) = \begin{cases} 1 - \frac{(m-3)4\pi}{R^2 b_0^4}[\frac{1}{s_1} - \frac{1}{s_2}] & \text{v=0} \\ \frac{(m-3)2\pi}{R^2 b_0^4 v}[e\sqrt{2\pi v} - \frac{1}{s_2}] & \ell_1 < v \le \ell_2 \\ \frac{(m-3)2\pi}{R^2 b_0^4 v}[(e-1)\sqrt{2\pi v}] & \ell_2 < v \le \ell_3 \\ \frac{(m-3)2\pi}{R^2 b_0^4 v}[\frac{1}{s_1} - \sqrt{2\pi v}] & \ell_3 < v \le \ell_4 \\ 0 & \text{otherwise} \end{cases}$$

Dropping n points convolves this distribution with itself $n-3$ times:

$$f(W_{\overline{H}_{m,n}} = v) = \bigotimes_{i=1}^{n-3} f(V_{\overline{H}_m})$$

For a model of size m and a correct hypothesis in an image with n points, the weight of the total vote for this hypothesis is the sum of weights over all $m-3$ other projected model points plus the sum of the weights of the $n-m$ clutter points. We will call this random variable $W_{H_{m,n}} = \sum_{i=1}^{m-3} V_{H_i} + \sum_{i=1}^{n-m} V_{\overline{H}_i}$. Though the random variables V_{H_i} are not independent, we make the simplifying assumption that they are, and proceed with the analysis. Assuming independence, the sum follows the distribution:

$$f(W_{H_{n,m}} = v) = \bigotimes_{i=1}^{m-3} f(V_{H_i}) \otimes \bigotimes_{i=1}^{n-m} f(V_{\overline{H}_i})$$

The validity of this assumption will be examined in a later section of this paper. We will use the central limit theorem to avoid actually having to compute this distribution, and will assume that the result of the convolution is Gaussian.

(g) Given these two distributions, we can now find the probability that an incorrect hypothesis will look like a correct one. The problem of deciding whether a sensor basis corresponds to a particular model basis is a simple binary hypothesis testing problem, for which we can easily find an optimum decision rule. We postpone the discussion of this rule until a later section; for now we will simply state that the decision rule yields a fixed probability of false positive (P_F) versus detection (P_D) as a function of threshold. It is also shown that this decision rule performs better for high clutter levels and degrades more gracefully as compared to the analogous optimal decision rule in the uniform bounded error case.

(h) Now let us step back and look at the overall decision problem. We pick three image points, and accumulate weights for $\binom{m}{3} * 6$ bases. Suppose we are willing to verify (by alignment or any other verification technique) all bases that pass the initial test, as long as there are $\le k$ of them. Then, an overall false positive is the combined event that the three image points being tested do not arise from the model, yet more than k model bases "look good". An overall true positive is the combined event that the three image points do arise from the model, that $\le k$ model bases pass the test, and of these, one of them is the correct one. We will call these combined events Ω_F and Ω_D, and

$$P(\Omega_F) = 1 - \sum_{i=0}^{k} P_F^i (1-P_F)^{\binom{m}{3}-i}$$

$$P(\Omega_D) = P_D * \sum_{i=0}^{k-1} P_F^i (1-P_F)^{\binom{m}{3}-i}$$

The following sections show the derivation of these distributions, and the results of the analysis both analytically and empirically.

5 Deriving the Projected Gaussian Distribution

In [GHJ91] an analytic expression for the case of circular error disks was derived as follows: given 3 model points (with model space coordinates) as basis, and the affine coordinates of a fourth model point with respect to this basis, the expression for the coordinates of the fourth point in model space is

$$\vec{m_4} = \vec{m_1} + \alpha(\vec{m_2} - \vec{m_1}) + \beta(\vec{m_3} - \vec{m_1}).$$

Under an arbitrary affine transformation T, each model point projects to the location

$$\vec{s_i} = T\vec{m_i} + \vec{e_i}$$

where $\vec{e_i}$ is a vector drawn from the error distribution. The possible location of the fourth model point is found by plugging the first expression into the second equation, to yield

$$\vec{s_4} = T\vec{m_4} + \vec{e_4}$$

where

$$\vec{e_4} = (1-\alpha-\beta)\vec{e_1} + \alpha\vec{e_2} + \beta\vec{e_3} + \vec{e_0}. \tag{1}$$

When the error vector is drawn from a uniform circular distribution with radius ϵ, the expression for the projected error vector is found to be

$$\epsilon[\mid 1-\alpha-\beta \mid + \mid \alpha \mid + \mid \beta \mid + 1] \tag{2}$$

For this paper, the sensor error vector is drawn from a two dimensional circular Gaussian distribution. The 2D Gaussian probability density of a random variable $\vec{a}$ with 0 covariance is denoted as:

$$\begin{aligned} f(\vec{a} = (x,y)) &= \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2}} \\ &= f(a_x = x) f(a_y = y) \end{aligned}$$

Because the two components are independent, the probability density of the sum of two random variables with $2D$ Gaussian distribution and 0 covariance is:

$$f(\vec{a} + \vec{b} = (x,y)) = f(a_x + b_x = x, a_y + b_y = y)$$

Convolution in each dimension yields:

$$\begin{aligned} &f(\vec{a} + \vec{b} = (x,y)) \\ &= \frac{1}{\sqrt{2\pi(\sigma_{ax}^2 + \sigma_{bx}^2)}} e^{-\frac{x^2}{2(\sigma_{ax}^2 + \sigma_{bx}^2)}} \\ &\cdot \frac{1}{\sqrt{2\pi(\sigma_{ay}^2 + \sigma_{by}^2)}} e^{-\frac{y^2}{2(\sigma_{ay}^2 + \sigma_{by}^2)}} \end{aligned}$$

Multiplying by a scalar yields:

$$\begin{aligned} f(c\vec{a} = (x,y)) &= f(a_x = \frac{x}{c}, a_y = \frac{y}{c}) \\ &= \frac{1}{\sqrt{2\pi}(c\sigma_x)} e^{-\frac{x^2}{2(c\sigma_x)^2}} \\ &\cdot \frac{1}{\sqrt{2\pi}(c\sigma_y)} e^{-\frac{y^2}{2(c\sigma_y)^2}} \end{aligned}$$

Therefore, assuming $\vec{e_i}$ to be $2D$ Gaussian with 0 covariance and standard deviations $\sigma_{ix} = \sigma_{jy} = \sigma$, the distribution of the vector in equation (1) is a $2D$ Gaussian with covariance 0 and standard deviation:

$$\sigma_e = \sigma((1-\alpha-\beta)^2 + \alpha^2 + \beta^2 + 1)^{1/2} \tag{3}$$

in both the x and y direction. Because the Gaussian distribution is not bounded, we choose to terminate the search for points after a radius of $2\sigma_e$, which means that we will find an image feature arising from a model point 86.5% of the time (this is demonstrated in a later section). Note that this expression is always smaller than its analogous expression for disk radius in the uniform bounded error model from equation (2) because of the triangle inequality. In the comparison, $\epsilon = 2\sigma$.

6 Determining the Distribution for σ_e

In the analysis we use two different probability densities for σ_e, one for correct basis matchings and one for incorrect basis matchings. Intuitively, this is due to the fact that when an incorrect basis matching is tested, more often than not the projected model points fall outside the image range and are thrown away, while when a correct hypothesis is tested the remaining model points always project to within the image. In tests we have observed that over half of the incorrect hypotheses are rejected for this reason, leading to an altered density for σ_e.

Let us call the two distributions $f_H(\sigma_e)$ and $f_{\overline{H}}(\sigma_e)$. We empirically estimate the former distribution by generating a random model of size 25, then for each ordered triple of model points as basis, we increment a histogram for the value of σ_e as a function of α and β for all the other model points with respect to that basis. For the latter distribution, we generate a random model of size 4 and a random image, and histogram the values of σ_e for only those cases in which the initial basis matching causes the remaining model point to fall within the image. The distributions for σ_e found in this manner have been observed to be invariant over many different values of model and image points.

The model is constrained such that the maximum distance between any two model points is not greater than 10 times the minimum distance, and in the basis selection, no basis is chosen such that the angle ψ between the two axes is $0 \leq \mid \psi \mid \leq \frac{\pi}{16}$ or $\frac{15}{16}\pi \leq \psi \leq \frac{17}{16}\pi$. This is done to avoid unstable bases.

The results were almost identical in every test we ran; two typical normalized histogram are shown in figure 1. For a choice of $\sigma = 2.5$, the histograms very closely fit the curves $f_H(\sigma_e) = (b_1\sigma_e)^{-2}$, $b_1 = 0.58$, and $f_{\overline{H}} = (b_2\sigma_e)^{-4}$, $b_2 = 0.35$ between the ranges $s_1 = 2.875$ and $s_2 = 120$. Figure 1 shows the estimated density functions shown superimposed on the empirical distributions. The integral of the analytic expression thus defined = 1.009 and 0.975, respectively.

7 Derivation of the Single Point Distributions

In this section we show the derivation of the distributions $f(V_H = v)$, the density function on the values that an image point contributes to a model basis given that the point comes from the model, and $f(V_{\overline{H}} = v)$, the density function on the values that an image point contributes to a basis given that it is a random point. We begin with the former.

7.1 Deriving $f(V_H)$

Given a correct hypothesis and no occlusion, the location of a projected model point can be modeled as a vector $\vec{\mathbf{d}}$ centered at the predicted location with Gaussian distribution (expressed in polar coordinates)

$$f(\vec{\mathbf{d}} = (r, \theta)) = \frac{1}{2\pi\sigma_e^2} e^{-\frac{r^2}{2\sigma_e^2}}$$

where we know σ_e and its distribution. We now choose an evaluation function $g(\vec{\mathbf{d}})$, which we use to weight a match that is offset by $\vec{\mathbf{d}}$ from the predicted match location. We want to find its density, *i.e.*, we want $f(v = g(\vec{\mathbf{d}}))$, where the distribution of $\vec{\mathbf{d}}$ is as stated. As mentioned, we choose the evaluation function

$$g(\vec{\mathbf{d}} = (r, \theta)) = \frac{1}{2\pi\sigma_e^2} e^{-\frac{r^2}{2\sigma_e^2}}$$

Since the evaluation function g is a really function of r alone, we need to know the density function of r. To find this, we integrate $f(r, \theta)$ over θ:

$$f(r) = \int_0^{2\pi} f(r, \theta) r d\theta = \frac{r}{\sigma_e^2} e^{\frac{-r^2}{2\sigma_e^2}}$$

Next, we want to find the density of the weight function $v = g(r)$. The change of variables formula for a monotonically decreasing function is:

$$\text{dens}(g(r) = v) = \frac{-f(g^{-1}(v))}{g'(g^{-1}(v))}$$

Working through the steps, we find

$$\begin{aligned}
g(r) &= \frac{1}{2\pi\sigma_e^2} e^{-\frac{r^2}{2\sigma_e^2}} \\
g'(r) &= -\frac{r}{\sigma_e^2} g(r) \\
f(r) &= \frac{r}{\sigma_e^2} e^{\frac{-r^2}{2\sigma_e^2}} \\
&= 2\pi r g(r) \\
f(g^{-1}(v)) &= 2\pi g^{-1}(v) g(g^{-1}(v)) \\
&= 2\pi v g^{-1}(v) \\
g'(g^{-1}(v)) &= -\frac{g^{-1}(v)}{\sigma_e^2} g(g^{-1}(v)) \\
&= -\frac{g^{-1}(v)}{\sigma_e^2} v \\
\Longrightarrow f(v = g(r)) &= \frac{-f(g^{-1}(v))}{g'(g^{-1}(v))} \\
&= 2\pi v g^{-1}(v) \frac{\sigma_e^2}{v g^{-1}(v)} \\
&= 2\pi\sigma_e^2
\end{aligned}$$

It may seem counterintuitive that the resulting distribution is constant. However, this can be understood if one considers an example in which $f(r, \theta)$ is uniformly distributed. Integrating over all angles yields a linearly increasing function in r. Assigning an evaluation function $g(\vec{\mathbf{d}})$ which is inversely proportional to r yields a constant density function on $f(v)$. The same thing is happening here, only quadratically. Since we only search for a match out to a radius of $2\sigma_e$, the effective distribution is:

$$f(V_H) = \begin{cases} \int_{u=g(\infty)}^{u=g(2\sigma_e)} 2\pi\sigma_e^2 du & \\ = e^{-2} & v = 0 \\ 2\pi\sigma_e^2 & \frac{1}{2\pi\sigma_e^2 e^2} \leq v \leq \frac{1}{2\pi\sigma_e^2} \\ 0 & \text{otherwise} \end{cases}$$

i.e., we will miss a good point $e^{-2} = 13.5\%$ of the time. This expression correctly integrates to 1. Now, note that in the expression we have a fixed σ_e, *i.e.*, we actually have derived $f(v = g(r) \mid \sigma_e)$. We need to integrate this expression over all values of σ_e, that is:

$$\begin{aligned}
f(V_H = v) &= \int f(V_H = v \mid \Sigma = \sigma) f_H(\Sigma = \sigma) d\sigma \\
&= \int (2\pi\sigma^2)(b_1\sigma)^{-2} d\sigma \\
&= \frac{2\pi}{b_1^2} \int d\sigma
\end{aligned}$$

There are two things to take into consideration when calculating the limits for this expression: first, the possible values of σ_e range from a lower limit s_1 to an upper limit s_2, due to limits on the values of the affine coordinates. (Earlier, we saw for $\sigma = 2.5$, that $s_1 = 2.875, s_2 = 120$). Also, for a given σ_e, it is clear that the maximum value we can achieve is when $r = 0 \Rightarrow v = \frac{1}{2\pi\sigma_e^2}$, and the minimum value we can achieve is at the cutoff point $r = 2\sigma_e \Rightarrow v = \frac{1}{2\pi\sigma_e^2} e^{-2}$. Setting v to each of these expressions and solving for σ_e leads to the conclusion that for a particular value v, the only values for σ_e such that $g(\vec{\mathbf{d}} \mid \sigma_e)$ could equal v are in the range $(\frac{1}{\sqrt{2\pi v} e}, \frac{1}{\sqrt{2\pi v}})$. Therefore the lower bound on the integral is $\sigma_e = \max(s_1, \frac{1}{\sqrt{2\pi v} e})$, and the upper bound is $\sigma_e = \min(\frac{1}{\sqrt{2\pi v}}, s_2)$. We split this integral into 3 regions, and deal with the case where $v = 0$ separately. Integrating, we get:

$$f(V_H = v) = \begin{cases} e^{-2} & \text{v=0} \\ \frac{2\pi}{b_1^2}(s_2 - \frac{1}{e\sqrt{2\pi v}}) & \ell_1 < v \leq \ell_2 \\ \frac{2\pi}{b_1^2} \frac{e-1}{e\sqrt{2\pi v}} & \ell_2 < v \leq \ell_3 \\ \frac{2\pi}{b_1^2}(\frac{1}{\sqrt{2\pi v}} - s_1) & \ell_3 < v \leq \ell_4 \\ 0 & \text{otherwise} \end{cases}$$

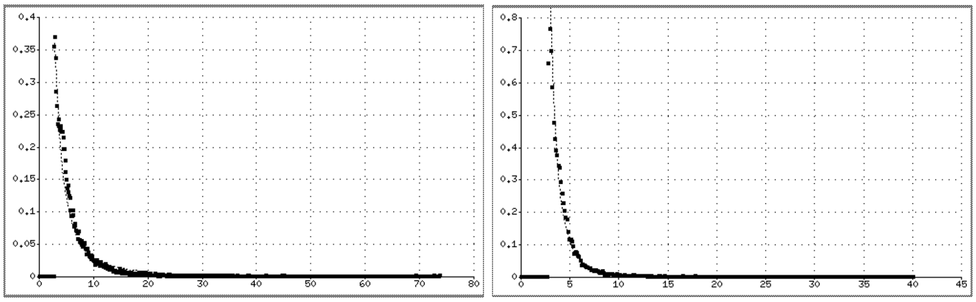

Figure 1: The distributions $f_H(\sigma_e)$ and $f_{\overline{H}}(\sigma_e)$.

where

$$\ell_1 = \frac{1}{2\pi {s_2}^2 e^2} \qquad \ell_2 = \frac{1}{2\pi {s_2}^2}$$
$$\ell_3 = \frac{1}{2\pi {s_1}^2 e^2} \qquad \ell_4 = \frac{1}{2\pi {s_1}^2}$$

7.1.1 Adding Occlusion

It is easy to add occlusion into this distribution by considering an independent process whose probability of occluding any given point is c. Therefore, the above distribution is multiplied by another factor:

$$f_c(V_H = v) = \begin{cases} f(V_H = 0)(1-c) + c & v = 0 \\ f(V_H = v)c & \text{otherwise} \end{cases}$$

We will use the distribution f, not f_c, in the rest of the paper, and will reconsider the rate of occlusion only in the context of calculating false negatives in a later section.

7.2 Deriving $f(V_{\overline{H}})$

We do the same derivation for the distribution $f(V_{\overline{H}})$. Given a hypothesis and a random point, we calculate the distribution as follows: let event A = "point falls in hypothesized error disk". This is the area of the error disk over the size of the image R^2, *i.e.*,

$$P(A \mid \sigma_e) = \frac{4\pi\sigma_e^2}{R^2}$$
$$P(\overline{A} \mid \sigma_e) = \frac{R^2 - 4\pi\sigma_e^2}{R^2}$$

Now we calculate the probability that a point which is uniformly distributed inside a disk of radius $2\sigma_e$ contributes value v for an incorrect hypothesis, using the evaluation function defined in the previous section. As before, we must express a uniform distribution in polar coordinates and then integrate over θ to get the distribution in terms of r alone, since the evaluation function g is a function of r:

$$f(r,\theta) = \frac{1}{\pi(2\sigma_e)^2}$$
$$f(r) = \int_0^{2\pi} \frac{1}{\pi(2\sigma_e)^2} r\,d\theta = \frac{r}{2\sigma_e^2}$$

As before, we calculate the density of $(v = g(r) \mid A, \sigma_e)$ with the new distribution for r and get:

$$f(v = g(r) \mid A) = \frac{-f(g^{-1}(v))}{g'(g^{-1}(v))} = \frac{1}{2}v^{-1}$$

Therefore, the density function of v for a fixed σ_e is:

$$f(V_{\overline{H}} \mid \sigma_e) = \begin{cases} P(\overline{A} \mid \sigma_e) \\ = \frac{R^2 - 4\pi\sigma_e^2}{R^2} & v = 0 \\ f(v \mid A, \sigma_e) P(A \mid \sigma_e) \\ = \frac{2\pi\sigma_e^2}{R^2 v} & \frac{1}{2\pi\sigma_e^2 e^2} \le v \le \frac{1}{2\pi\sigma_e^2} \\ 0 & \text{otherwise} \end{cases}$$

Again, this expression correctly integrates to 1. As before, we need to integrate over all values of σ_e:

$$f(V_{\overline{H}} = v) = \int f(V_H = v \mid \Sigma = \sigma) f_{\overline{H}}(\Sigma = \sigma) d\sigma = \int (\frac{2\pi\sigma^2}{R^2 v})(b_0\sigma)^{-4} d\sigma = \frac{2\pi}{b_0^4 R^2 v} \int \sigma^{-2} d\sigma$$

Dealing with $v = 0$ as a separate case, and with the same bounds as before, integrating yields:

$$f(V_{\overline{H}}) = \begin{cases} 1 - \frac{4\pi}{R^2 b_0^4}[\frac{1}{s_1} - \frac{1}{s_2}] & \text{v=0} \\ \frac{2\pi}{R^2 b_0^4 v}(e\sqrt{2\pi v} - \frac{1}{s_2}) & \ell_1 < v \le \ell_2 \\ \frac{2\pi}{R^2 b_0^4 v}(e-1)\sqrt{2\pi v} & \ell_2 < v \le \ell_3 \\ \frac{2\pi}{R^2 b_0^4 v}(\frac{1}{s_1} - \sqrt{2\pi v}) & \ell_3 < v \le \ell_4 \\ 0 & \text{otherwise} \end{cases}$$

where

$$\ell_1 = \frac{1}{2\pi {s_2}^2 e^2} \qquad \ell_2 = \frac{1}{2\pi {s_2}^2}$$
$$\ell_3 = \frac{1}{2\pi {s_1}^2 e^2} \qquad \ell_4 = \frac{1}{2\pi {s_1}^2}$$

We ran an experiment to test the analysis of this section, and the results are shown in Figure 2. Both graphs show a normalized histogram of the results of 15,000 independent trials. The first graph indicates the empirical results corroborating the predictions very closely. While the comparison of the second graph is less visually striking, note that the deviation at any point between the empirical and predicted results is generally less than one count.

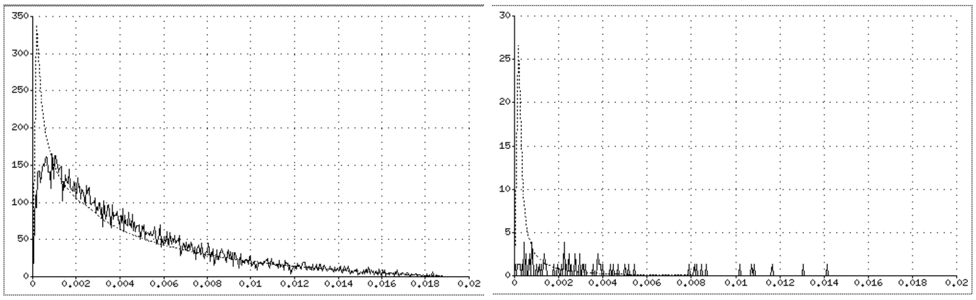

Figure 2: Distributions, $f(v)$, with and without model

8 Finding the Weight Density of a Model in an Image

Having found the single point densities, we use them to find the density of the combined weight of points for correct and incorrect hypotheses. We start with the density function on weights of correct hypotheses. For a model of size m and an image of size n, a correct hypothesis should have weight density

$$f(W_{H_m} = v) = \bigotimes_{i=1}^{m-3} f(V_{H_i}) \otimes \bigotimes_{i=1}^{n-m} f(V_{\overline{H}_i})$$

assuming that each point contributes weight to its supporting basis independently of any other. In order to avoid convolving the distributions from the previous section, we find the expected value and the standard deviation of the distributions and invoke the central limit theorem to claim that the combined weight of a correct hypothesis of a size m model in a size n image with should roughly follow the distribution:

$$N(mE_H + (n-m)E_{\overline{H}}, m\sigma_H^2 + (n-m)\sigma_{\overline{H}}^2)$$

in which

$$\begin{aligned}
E_H(v) &= \int_0^0 v f_c(v)dv + \int_{\ell_1}^{\ell_2} v f_c(v)dv \\
&\quad + \int_{\ell_2}^{\ell_3} v f_c(v)dv + \int_{\ell_3}^{\ell_4} v f_c(v)dv \\
&= (1-c)\frac{e^4-1}{12\pi b_1^2 e^4}\left[\frac{1}{s_1^3} - \frac{1}{s_2^3}\right] \\
&= 2.604 \times 10^{-2} \times \frac{(1-c)}{b_1^2}\left[\frac{1}{s_1^3} - \frac{1}{s_2^3}\right]
\end{aligned}$$

$$\begin{aligned}
E_H(v^2) &= \int_0^0 v^2 f_c(v)dv + \int_{\ell_1}^{\ell_2} v^2 f_c(v)dv \\
&\quad + \int_{\ell_2}^{\ell_3} v^2 f_c(v)dv + \int_{\ell_3}^{\ell_4} v^2 f_c(v)dv \\
&= (1-c)\frac{e^6-1}{60\pi^2 b_1^2 e^6}\left[\frac{1}{s_1^5} - \frac{1}{s_2^5}\right] \\
&= 1.6845 \times 10^{-3} \times \frac{(1-c)}{b_1^2}\left[\frac{1}{s_1^5} - \frac{1}{s_2^5}\right] \\
\sigma_H^2 &= E_H(v)^2 - E_H(v^2)
\end{aligned}$$

For an incorrect hypothesis we look at the problem in two steps. First we derive, as above, the mean and standard deviation of the process in which $n = m = 4$, *i.e.*, a single random image point drops into a single error circle. From the distribution of $f(V_{\overline{H}})$, we find:

$$\begin{aligned}
E_{\overline{H}}(v) &= \int_0^0 v f(V_{\overline{H}})dv + \int_{\ell_1}^{\ell_2} v f(V_{\overline{H}})dv \\
&\quad + \int_{\ell_2}^{\ell_3} v f(V_{\overline{H}})dv + \int_{\ell_3}^{\ell_4} v f(V_{\overline{H}})dv \\
&= \frac{(e^2-1)}{3e^2R^2b_0^4}\left[\frac{1}{s_1^3} - \frac{1}{s_2^3}\right] \\
&= .2882 \times \frac{1}{b_0^4R^2}\left[\frac{1}{s_1^3} - \frac{1}{s_2^3}\right]
\end{aligned}$$

$$\begin{aligned}
E_{\overline{H}}(v^2) &= \int_0^0 v^2 f(V_{\overline{H}})dv + \int_{\ell_1}^{\ell_2} v^2 f(V_{\overline{H}})dv \\
&\quad + \int_{\ell_2}^{\ell_3} v^2 f(V_{\overline{H}})dv + \int_{\ell_3}^{\ell_4} v^2 f(V_{\overline{H}})dv \\
&= \frac{(e^4-1)}{20e^4R^2b_0^4\pi}\left[\frac{1}{s_1^5} - \frac{1}{s_2^5}\right] \\
&= 1.554 \times 10^{-2} \times \frac{1}{b_0^4R^2}\left[\frac{1}{s_1^5} - \frac{1}{s_2^5}\right] \\
\sigma_{\overline{H}}^2(v) &= E_{\overline{H}}(v)^2 - E_{\overline{H}}(v^2)
\end{aligned}$$

Plugging in the values $s_1 = 2.875, s_2 = 120, b_0 = 0.35, b_1 = 0.58, c = 0$, and $R = 500$ for the experimental data of section 6 yields

$$\begin{aligned}
E_H &= 3.26 \times 10^{-3} \\
\sigma_H &= 1.49 \times 10^{-5} \\
E_{\overline{H}} &= 3.19 \times 10^{-6} \\
\sigma_{\overline{H}}^2 &= 2.08 \times 10^{-8}
\end{aligned}$$

Note that the value of the limit s_2 was determined empirically and is a function of the constraints on the bases that are chosen. Without the basis constraints, s_2 tends to infinity, and in fact the values of these parameters for $s_2 = 120$ and $s_2 = \infty$ are not significantly different.

Now, consider a single random image point (*i.e.*, $n = 4$; three for the hypothesis and one left over) dropped into an image where a model of size $m > 4$ is hypothesized to be. In this case the event that the random point will contribute weight v to this hypothesis is calculated as follows: Let event A_i = "point drops in the ith circle." Then,

$$\begin{aligned} & f(V_{\overline{H}_m} = v \mid v \neq 0) \\ &= f(v, A_1) + f(v, A_2) + \ldots + f(v, A_{m-3}) \\ &= (m-3) f(v, A_1) \end{aligned}$$

Note that because we are assuming the circles are disjoint, we are overestimating the probability of the point falling in any circle. The actual rate of detection will be lower than our assumption, especially as the m grows large.

$$f(V_{\overline{H}_m} = v) = \begin{cases} 1 - \frac{(m-3)4\pi}{R^2 b_0^4}[\frac{1}{s_1} - \frac{1}{s_2}] & \text{v=0} \\ \frac{(m-3)2\pi}{R^2 b_0^4 v}[e\sqrt{2\pi v} - \frac{1}{s_2}] & \ell_1 < v \leq \ell_2 \\ \frac{(m-3)2\pi}{R^2 b_0^4 v}[(e-1)\sqrt{2\pi v}] & \ell_2 < v \leq \ell_3 \\ \frac{(m-3)2\pi}{R^2 b_0^4 v}[\frac{1}{s_1} - \sqrt{2\pi v}] & \ell_3 < v \leq \ell_4 \\ 0 & \text{otherwise} \end{cases}$$

As m grows large, $(1 - (m-3)\frac{4\pi}{R^2 b_0^4}[s_2 - s_1]) < 0$ so this expression is no longer a density function. This is the point at which the model covers so much of the image that a random point will *always* contribute to some incorrect hypothesis. Therefore, this analysis only applies to models for which which $m < \frac{R^2 b_0^4}{r\pi[s_2 - s_1]} + 3$. For $R = 500, m \leq 60$, and for $R = 256, m \leq 18$.

The mean and standard deviation for one random point dropping into $m-3$ random circles is:

$$\begin{aligned} E_{\overline{H}_m}(v) &= \int_0^0 v f(V_{\overline{H}_m}) dv + \int_{\ell_1}^{\ell_2} v f(V_{\overline{H}_m}) dv \\ &\quad + \int_{\ell_2}^{\ell_3} v f(V_{\overline{H}_m}) dv + \int_{\ell_3}^{\ell_4} v f(V_{\overline{H}_m}) dv \\ &= 0 + \int_{\ell_1}^{\ell_2} v[(m-3) f(V_{\overline{H}})] dv \\ &\quad + \int_{\ell_2}^{\ell_3} v[(m-3) f(V_{\overline{H}})] dv \\ &\quad + \int_{\ell_3}^{\ell_4} v[(m-3) f(V_{\overline{H}_m})] dv \\ &= (m-3) E_{\overline{H}}(v) \\ E_{\overline{H}_m}(v^2) &= \int_0^0 v^2 f(V_{\overline{H}_m}) dv + \int_{\ell_1}^{\ell_2} v^2 f(V_{\overline{H}_m}) dv \\ &\quad + \int_{\ell_2}^{\ell_3} v^2 f(V_{\overline{H}_m}) dv + \int_{\ell_3}^{\ell_4} v^2 f(V_{\overline{H}_m}) dv \\ &= 0 + \int_{\ell_1}^{\ell_2} v^2[(m-3) f(V_{\overline{H}})] dv \\ &\quad + \int_{\ell_2}^{\ell_3} v^2[(m-3) f(V_{\overline{H}})] dv \\ &\quad + \int_{\ell_3}^{\ell_4} v^2[(m-3) f(V_{\overline{H}_m})] dv \\ &= (m-3) E_{\overline{H}}(v^2) \\ \sigma^2_{\overline{H}_m}(v) &= E_{\overline{H}_m}(v)^2 - E_{\overline{H}_m}(v^2) \\ &= (m-3)^2 E_{\overline{H}}(v)^2 - (m-3) E_{\overline{H}}(v^2) \end{aligned}$$

Dropping n points convolves this distribution with itself $n-3$ times:

$$f(W_{\overline{H}_{m,n}} = v) = \bigotimes_{i=1}^{n-3} f(V_{\overline{H}_m})$$

And therefore the weight that an n-size random image contributes to an incorrectly hypothesized model of size m follows the distribution:

$$N((n-3) E_{\overline{H}_m}, (n-3)\sigma^2_{\overline{H}_m})$$

Note that this is the weight density of a single incorrect hypothesis.

The means for both distributions were tested empirically from the same experiment as shown in Figures 2. A table of values is given in figure 3.

9 Interpreting the Results

We have derived expressions for the weight densities of a hypothesis given that it is incorrect, and given that it is correct. We are interested in using these distributions to determine the effectiveness of geometric hashing under different clutter conditions. To do this, we briefly introduce the ROC (receiver operating characteristic) curve, a concept borrowed from standard hypothesis testing theory, and cast our problem in terms of this framework.

9.1 ROC: Introduction

The problem is to decide which one of two hypotheses, H_0 and H_1, is correct. There is a random variable whose distribution is known given one or the other hypothesis, *i.e.*, we know $f(X \mid H_0)$ and $f(X \mid H_1)$. Let the space of all possible values of the random variable X be divided into two regions, Z_0 and Z_1, such that we decide H_0 if the value of X falls in Z_0 and H_1 if X falls in Z_1. Then we can define the quantities

$$\begin{aligned} & Pr(\text{say } H_0 \mid H_0 \text{ is true}) = \int_{Z_0} p(X \mid H_0) dX \\ P_F &= Pr(\text{say } H_1 \mid H_0 \text{ is true}) = \int_{Z_1} p(X \mid H_0) dX \\ P_M &= Pr(\text{say } H_0 \mid H_1 \text{ is true}) = \int_{Z_0} p(X \mid H_1) dX \\ P_D &= Pr(\text{say } H_1 \mid H_1 \text{ is true}) = \int_{Z_1} p(X \mid H_1) dX \end{aligned}$$

These quantities are often referred to as P_M= "Probability of a miss", P_D= "Probability of detection", and P_F= "Probability of false alarm" for historical reasons.

One way of constructing a decision rule is to use the likelihood ratio test (LRT) to divide the observation space into decision regions, i.e.,

	Mean			Variance		
With M	Empirical	Predicted	Emp/Pred	Empirical	Predicted	Emp/Pred
m-3=1,n-3=1	3.6953E-3	3.2177E-3	1.148	1.5186E-5	1.4625E-5	1.038
m-3=1,n-3=100	3.8383E-3	3.5339E-3	1.086	1.7350E-5	1.6680E-5	1.040
m-3=1,n-3=500	4.8026E-3	4.8115E-3	.9981	2.2274E-5	2.4984E-5	.8915
m-3=5,n-3=5	1.9658E-2	1.6089E-2	1.222	1.4927E-4	7.3124E-5	2.041
m-3=10,n-3=10	4.1986E-2	3.2177E-2	1.305	5.4130E-4	1.4625E-4	3.701
m-3=10,n-3=100	4.4513E-2	3.5052E-2	1.270	5.3400E-4	1.6485E-4	3.239
m-3=10,n-3=500	5.5476E-2	4.7828E-2	1.160	5.7484E-4	2.4752E-4	2.322

	Mean			Variance		
Without M	Empirical	Predicted	Emp/Pred	Emp	Predicted	Emp/Pred
m-3=1,n-3=1	3.2410E-6	3.1940E-6	1.015	1.8747E-8	2.0760E-8	.8897
m-3=1,n-3=100	3.0681E-4	3.1940E-4	.9606	1.9738E-6	2.0760E-6	.9508
m-3=1,n-3=500	1.6344E-3	1.5970E-3	1.023	1.1163E-5	1.0380E-5	1.075
m-3=5,n-3=5	8.9131E-5	7.9850E-5	1.116	6.4808E-7	5.1797E-7	1.251
m-3=10,n-3=10	3.4949E-4	3.1940E-4	1.094	2.4001E-6	2.0668E-6	1.161
m-3=10,n-3=100	3.5082E-3	3.1940E-3	1.098	2.3277E-5	2.0668E-5	1.126
m-3=10,n-3=500	1.6289E-2	1.5970E-2	1.020	1.0766E-4	1.0334E-4	1.042

Figure 3: A table of predicted versus empirical means and variances of the distribution $f(W_{H_{m,n}} = v)$, in the top table, and $f(W_{\overline{H}_{m,n}} = v)$ in the bottom table, for different values of m and n.

$$\frac{p(X \mid H_1)}{p(X \mid H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \eta$$

That is, if the ratio of the conditional densities is greater than a fixed threshold η, choose H_1, otherwise choose H_0. Note that changing the value of η changes the decision regions and thus the values of $P_F$and P_D. The ROC curve is simply the graph of $P_D$versus $P_F$as a function of threshold for the LRT. As it turns out, both the Neyman-Pearson test and the optimal Bayes test involve this LRT, thus the ROC curve encapsulates all information needed for either test, since any (P_F, P_D) point yielded by either test necessarily lies on the ROC curve. If the prior probabilities of H_0 and H_1 are known, then the optimal Bayes decision rule picks the ROC point which minimizes the expected cost of the decision by using the LRT in which the threshold is a function of the costs and priors involved:

$$\eta = \frac{(C_{10} - C_{00})P_0}{(C_{01} - C_{11})P_1}$$

where $C_{i,j}$ is the cost associated with choosing hypothesis i given that hypothesis j is correct. In the absence of such priors, a Neymann Pearson test is often considered optimal, in which one simply picks a point on the ROC curve which gives satisfactory performance. Note that this is *not* the same as minimizing the decision's expected cost.

For example, assume for our problem that $H_0 \sim N(m_0, \sigma_0^2)$ and $H_1 \sim N(m_1, \sigma_1^2)$, and assume that $m_1 > m_0$ and $\sigma_1 > \sigma_0$. The likelihood ratio test yields:

$$\left(\frac{X - m_0}{\sigma_0}\right)^2 - \left(\frac{X - m_1}{\sigma_1}\right)^2 \underset{H_0}{\overset{H_1}{\gtrless}} 2\ln\frac{\eta\sigma_1}{\sigma_0} = \gamma$$

The regions Z_0 and Z_1 are found by solving the above equation for equality,

$$X_1 = \frac{[(m_1\sigma_0^2 - m_0\sigma_1^2) - \sigma_0\sigma_1(\gamma[\sigma_1^2 - \sigma_0^2] + (m_0 - m_1))^{1/2}]}{\sigma_1^2 - \sigma_0^2}$$
$$X_2 = \frac{[(m_1\sigma_0^2 - m_0\sigma_1^2) - \sigma_0\sigma_1(\gamma[\sigma_1^2 - \sigma_0^2] + (m_0 - m_1))^{1/2}]}{\sigma_1^2 - \sigma_0^2}$$

The values of $P_F$and $P_D$are found by integrating the conditional probability densities $p(X \mid H_0)$ and $p(X \mid H_1)$ over these regions Z_0 and Z_1:

$$P_F = \int_{Z_1} p(X \mid H_0)dX = 1 - \int_{X_1}^{X_2} \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{x-m_0^2}{2\sigma_0^2}}$$
$$P_D = \int_{Z_1} p(X \mid H_1)dX = 1 - \int_{X_1}^{X_2} \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{x-m_1^2}{2\sigma_1^2}}$$

In figure 4 for example, we have plotted the ROC curve for the distributions $f(X \mid H_0)$ and $f(X \mid H_1)$ alongside. The axes are $x = P_F$, $y = P_D$. The line $x = y$ is a lower bound, since a points on this line indicate that any decision is as likely to be true as false, so the observed value of X gives us no information. Though an ROC curve is a 3D entity (i.e., a point in (P_F, P_D, η) space), we display its projection onto the $\eta = 0$ plane and can easily find the associated η value for any (P_F, P_D) pair. When the threshold is high there is a 0 probability of false negative, but a 0 probability of correct identification as well. As the threshold goes down, the probabilities of both occurences go up until the threshold is so low that both positive and false identification are certain. In our problem we assume that we do not have priors, so our goal is to pick a threshold such that we have a very high probability of identification and a

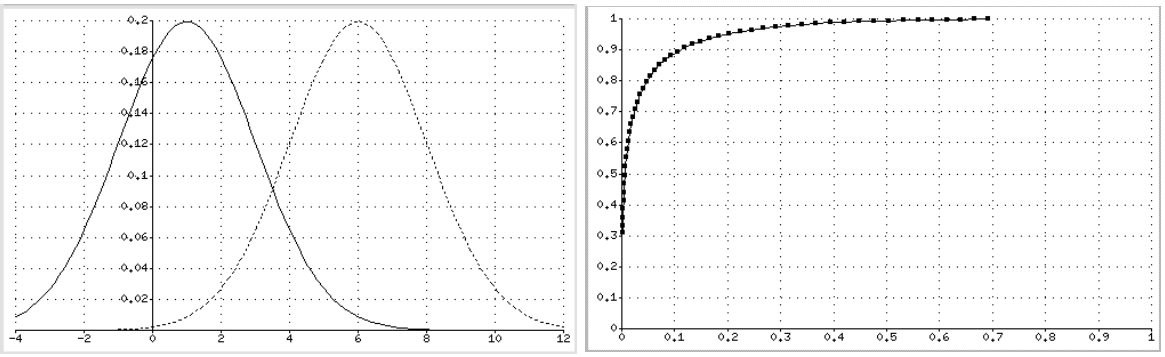

Figure 4: On the left is displayed the conditional probability density functions of a random variable X. On the right is the associated ROC curve, where $P_F$and $P_D$correspond to the x and y axes, respectively.

low probability of false positives, *i.e.*, we are interested in picking a point as close to the upper left hand side as possible. Note that the larger the separation between the two hypothesis distributions, the more the curve is pushed towards that direction.

9.2 Applying ROC to Geometric Hashing

In our problem formulation, H_0 = probability that the model is not in the image, and H_1 = probability that it is. In our case, we have a different ROC curve associated with every fixed (m, n) pair, where m and n are the number of model and image features, respectively.

The next examples show the predicted comparison of the Gaussian model to the bounded uniform model. Figure 5 shows the ROC curves for the Gaussian and uniform models, $m - 3 = 10$,$n - 3 = 10, 50, 100, 500, 1000$, occlusion=0.0 and 0.25. We can see that in the case of no occlusion, for small values of n, both models predict good $P_F$vs $P_D$curves, though the bounded uniform model will always be better because there is no possibility of a false negative for occlusion=0, while in the unbounded Gaussian case there always is. However, as n increases, the uniform model breaks down more rapidly than the Gaussian model for both occlusion values. For occlusion=0.25, both models perform about equally for small values of n (for example, at $n = 100$), but again as n increases, the uniform error model fails more dramatically than the Gaussian model ($n \geq 500$).

Using this technique, we can predict thresholds for actual experiments, as shown in the next section.

10 Experiment

The predictions of the previous section were tested in the following experiment: to test an ROC curve for model size m, image size n, we run two sets of trials, one to test the probability of detection and one to test the probability of false alarm. For the former, a random model of size m consisting of point features was generated and projected into an image, with Gaussian noise ($\sigma = 2.5$) added to both the x and y positional components of each point feature. Occlusion (c) is simulated by adding a c probability of not appearing in the resulting image for each point. Only correct correspondences are tested, and the weight of each of these correct hypotheses is found using the algorithm:

(a) for a correct hypothesis $(m_0 : i_0; m_1 : i_1; m_2 : i_2)$

for every other model point m_j

(i) find coordinates $m_j = (\alpha_j, \beta_j)$ with respect to basis (m_0, m_1, m_2), and from this, $\sigma_e = f(\alpha_j, \beta_j)$.

(ii) For every image point i_j, find the minimum distance d between i_j and any of the projected points such that $d \leq 2\sigma_e$. Add $v = \frac{1}{2\pi\sigma_e^2} e^{-\frac{d^2}{2\sigma_e^2}}$ to the supporting weight for this hypothesis.

(b) If the weight of the vote for this hypothesis is greater than some threshold θ, stop and output this as a correct instance of the model.

For our experiment, we loop through thresholds from 0 to $E_H(v)$, and for every threshold we run the above algorithm enough times to get 100 sample points. To test the probability of false alarm, we run the same experiment exactly, except we use random images which do not contain the model we are looking for. We loop through the same thresholds as in the previous case to get a set of (P_F,P_D) pairs for each threshold. The resulting P_F, P_D, and ROC curves are shown in figure 6 for $n - 3 = 10, 100, 500, 500$, occlusion $c = 0.0, 0.0, 0.0, 0.25$. The ROC curves for the same parameters are shown alongside.

In the cases of no occlusion, the predicted and empirical curves match very nicely. However, for occlusion=0.25, the empirical ROC curve falls below our expectations. This is due to the fact that the distribution of W_H has a larger variance than our predicted value (see table 3 and figure 7). In fact, though we assumed at the outset of the analysis that the individual random variables V_H were independent, this is not the case; for a correct basis matching, the joint distribution of any two error vectors $\vec{e}_i, \vec{e}_j, i, j \neq 0, 1, 2$ can shown to have a non-zero covariance:

$$\Lambda_{i,j} = (1 - \alpha_i - \beta_i)(1 - \alpha_j - \beta_j) + \alpha_i\alpha_j + \beta_i\beta_j$$

This leads to a larger variance for the overall distribution than that predicted using the independence as-

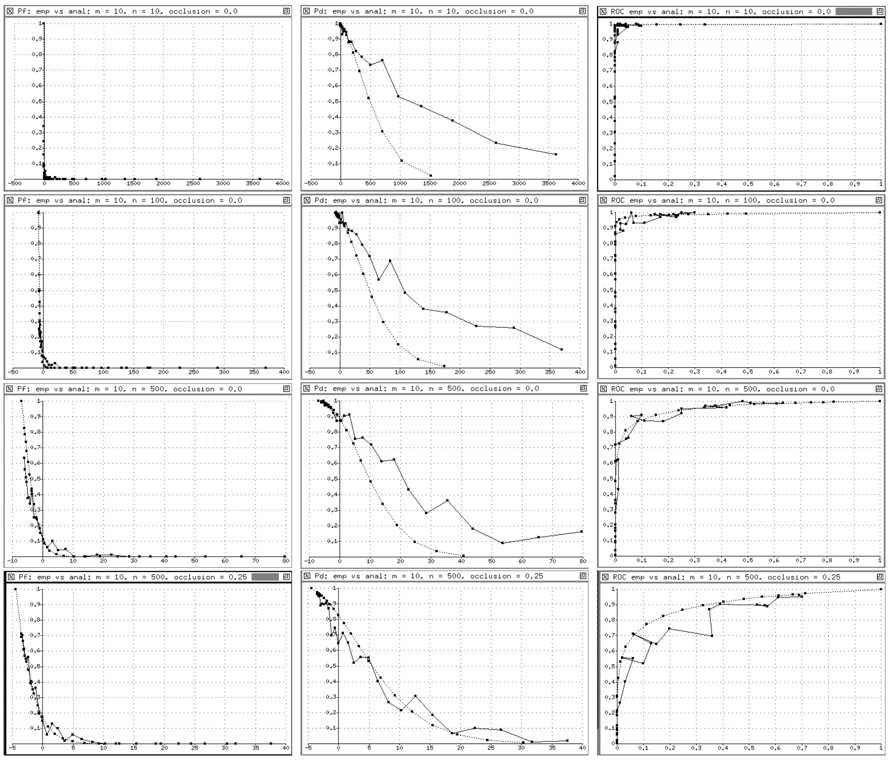

Figure 6: Comparison of predicted to empirical curves for probability of false alarm, probability of detection, ROC curves. From top to bottom, $n - 3 = 10, 100, 500, 500$, occlusion $= 0, 0, 0, 025$.

sumption and hence poorer results. We are currently working on another analysis that takes this dependence into account.

11 Conclusion

The geometric hashing method was introduced by Lamdan, Schwartz and Wolfson in 1987. The first error analysis of the geometric hashing technique was done by Grimson, Huttenlocher and Jacobs, who showed that with even very small amounts of noise and spurious features, the technique had a very high probability of false positives. However, they assumed that the error was uniform and bounded, which is a worst-case scenario and places an upper bound on the error rate. As we have shown here, with a Gaussian error assumption we can do much better.

Costa, Haralick, and Shapiro demonstrated another error analysis for geometric hashing [CHS90] also based on a 2D Gaussian noise distribution associated with each point. Their analysis differs from this one technically in many respects, but the main difference is that they assume that the model they are looking for is present in the image and they focus on finding the pose by deriving an optimal voting scheme. This is in contrast to the work presented here, in which given a voting scheme and no prior information about the presence or absence of the model, we explicitly derived the probability of false detection as a function of clutter, and characterized the confidence level of the hypotheses that the method offers as "correct". We did this by choosing a hypothesis evaluation function, and deriving the probability density of the evaluation function on both correct and incorrect hypotheses to determine, when given some hypothesis, which distribution it was drawn from. We showed also that the Gaussian error model separates the two distributions more than the uniform bounded error model, leading to better ROC curves.

The contribution of this work is to cast the geometric hashing technique in terms of standard estimation theory, which has several advantages. The ROC curve formulation explicitly demonstrates the performance achievable for a given signal to noise ratio as a function of acceptance threshold. Given a desired detection rate, the user can determine from the ROC curve what acceptance threshold to use in order to minimize the probability of false detections. In this formulation it is also clear when adequate performance cannot be achieved, for if the desired minimum performance point (P_F, P_D) lies above the ROC curve for a particular clutter level, then this performance is not possible no matter what operating parameters are chosen. The ROC formulation is also a succinct method for comparing voting schemes, as we compared the voting schemes implied by the Gaussian versus uniform error models. We expect to be able to use such techniques to choose thresholds analytically instead of heuristically in recognition systems.

12 Acknowledgements

I would like to thank David Jacobs and Sandy Wells for helpful discussions on this topic. Particular thanks goes

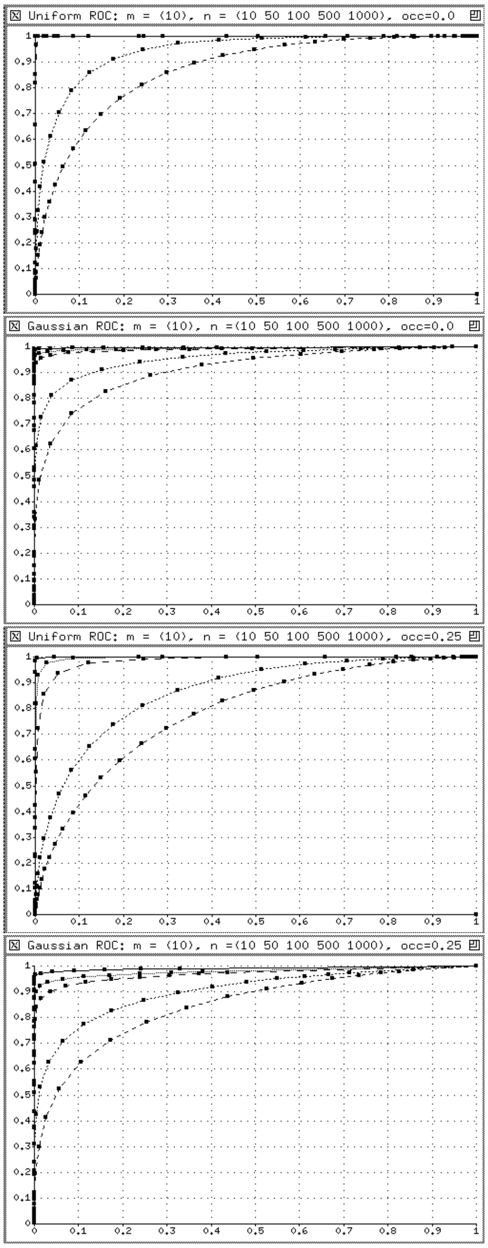

Figure 5: Comparisons of uniform and Gaussian error models for $m-3=10$, $n-3=10,50,100,500,1000$. From top: uniform, occlusion=0; Gaussian, occlusion=0; uniform, occlusion=.25, Gaussian, occlusion=.25.

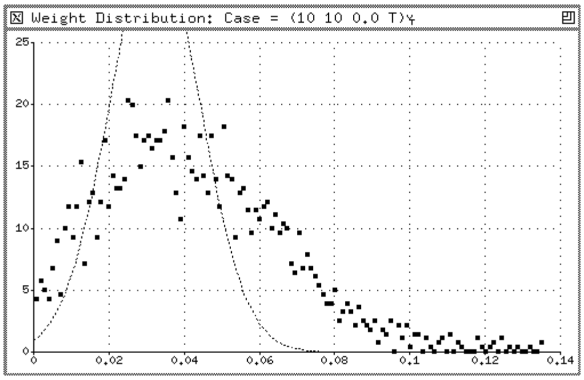

Figure 7: Comparison of predicted to empirical distribution for $W_{H_{10,10}}$, occlusion = 0. Note that the actual distribution has much greater variance than predicted.

to Eric Grimson for comments on organization and presentation, and for his suggestion of the original problem.

References

[CHS90] M. Costa, R.M. Haralick, and L.G. Shapiro. Optimal Affine-Invariant Point Matching. In *Proc. 6th Israel Conf. Art. Intell.*, pages 35–61, 1990.

[GHJ91] W.E.L. Grimson, D.P. Huttenlocher, and D.W. Jacobs. Affine Matching with Bounded Sensor Error: A Study of Geometric Hashing and Alignment. Technical Report 1250, M.I.T. AI Lab, 1991.

[HW88] R. Hummel and H. Wolfson. Affine Invariant Matching. In *DARPA Image Understanding Workshop*, 1988.

[LSW87] Y. Lamdan, J.T. Schwartz, and H.J. Wolfson. On Recognition of 3-D Objects from 2-D Images. Technical Report 322, N.Y.U. Dept. Comp. Sci., 1987.

[RH91] I. Rigoutsos and R. Hummel. Robust Similarity Invariant Matching in the Presence of Noise. In *8th Israeli Conf. Art. Intell. Comp. Vis.*, 1991.

[VT68] Harry L. Van Trees. *Detection, Estimation, and Modulation Theory*, volume I: Detection, Estimation, and Linear Modulation Theory, chapter 2. John Wiley and Sons, 1968.

[Wel91] William Wells. MAP Model Matching. In *IEEE Comp. Vis. Patt. Recog.*, pages 486–492, 1991.