Error Propagation in Full 3D-from-2D Object Recognition

T. D. Alter
MIT AI Laboratory[1]
545 Technology Square, Room 751
Cambridge, MA 02139
tda@ai.mit.edu

David W. Jacobs
NEC Research Institute
4 Independence Way
Princeton, NJ 08540
dwj@research.nj.nec.com

Abstract

Robust recognition systems require a careful understanding of the effects of error in sensed features. Error in these image features results in uncertainty in the possible image location of each additional model feature. We present an accurate, analytic approximation for this uncertainty when model poses are based on matching three image and model points. This result applies to objects that are fully three-dimensional, where past results considered only two-dimensional objects. Further, we introduce a linear programming algorithm to compute this uncertainty when poses are based on any number of initial matches.

1 Introduction

Object recognition systems frequently hypothesize a known object's pose based on matching a small number of the object's features to features in the image (e.g., [7, 14, 17, 10]). To confirm the hypothesis, they commonly use the pose to look for additional matches. A fundamental question in building robust recognition systems is how noise in the matched image features can propagate to uncertainty in the locations of still unmatched object features.

Here we consider point features. We model error by assuming that a detected feature point is no more than ϵ pixels from the location of the true point, without committing to a particular error distribution (although see [12] for a treatment of Gaussian error). This assumption is valid when, with high probability, the error in the image points can be limited to small areas. Also, this assumption is useful for limiting the region to search for confirming features. Given a set of matched image and model points, and an unmatched model point, we show how to accurately compute the *uncertainty region* where that model point might appear, assuming scaled orthographic (weak-perspective) projection.

Many early recognition systems took an ad-hoc approach to accounting for the effects of sensing error on the projected positions of unmatched model features. Recently there has been much effort directed at better understanding the effects of error on the matching process. For 3D models and 3D data, [6] computed bounds on the model pose given a set of matches between model and image features. For recognition of planar objects from 2D images, [11, 16] considered the effects of error on alignment approaches and [5, 15, 9, 13, 16] examined the effects of error on the related Geometric Hashing approach.

For full 3D objects, [8] considered error in the case of non-linear, scaled orthographic projection of 3D objects into 2D images. They provided a numerical method for bounding the uncertainty regions of points, based on a match between three model and image points. This result is useful for analyzing the performance of alignment and transformation clustering approaches to recognition, but is less practical for use at run-time in a recognition system. [1] presented experiments that show that the true uncertainty regions tend to be circular to a good approximation, and presented another numeric method for more accurately bounding the uncertainty regions.

We derive a simple linear expression that approximates the relationship between the matched and unmatched points. Then we show empirically that this expression is accurate for the amount of error that is of interest in most recognition applications. This relationship allows us to show that when we have matched three points, the uncertainty region for a fourth point is approximately circular, and to derive analytic expressions for the center and radius of the circle. Similarly, when there is Gaussian error in the image points, this relationship implies that the propagated distribution of uncertainty is also Gaussian, with analytic

[1] TDA was supported by Office of Naval Research (AASERT) Reference No. N00014-94-1-0128 and by NEC Research Insitute.

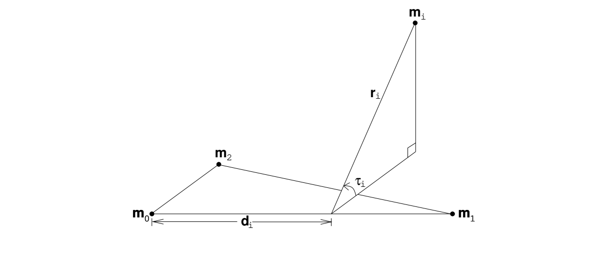

Figure 1: Every model point has unique (d, r, τ) coordinates, unless it is on the line through $\vec{m_{01}}$, where τ is unrestricted. Note $\tau = 0$ for points in the basis plane.

expressions for the center and standard deviation.

We also take advantage of this linear relationship by introducing a new algorithm that allows us to determine the uncertainty region for any number of matched points. To do this we approximate our bounded error regions with convex polygons, and then show that we can use linear programming to derive a convex polygon that describes the uncertainty region of the unmatched model point.

2 4th-Point Uncertainty Region

In this section, we are interested in the following problem: Given three matching pairs of image and model points, $(\vec{i_0}, \vec{m_0})$, $(\vec{i_1}, \vec{m_1})$, and $(\vec{i_2}, \vec{m_2})$, where the locations of image points $\vec{i_0}$, $\vec{i_1}$, and $\vec{i_2}$ contain a small amount of error, what is the error in the computed image position of a fourth model point $\vec{m_3}$? This section presents an analytic solution to this problem, which is based on a first-order approximation, and results in a linear relationship between the errors in the matched (basis) image points and the error in any unmatched, projected model point. We consider only weak-perspective projection. In [12], we apply the technique to perspective projection to obtain again a linear relationship between the errors, but in this case the relationship is much more complicated.

We assume for now that the image positions of $\vec{m_0}$, $\vec{m_1}$, and $\vec{m_2}$ equal $\vec{i_0}$, $\vec{i_1}$, and $\vec{i_2}$, respectively. We use the following representation of 3D model points in terms of the three basis model points (see Fig. 1): Originally, the model points lie in some model coordinate system. For any model point $\vec{m_3}$, let r be length of the perpendicular from $\vec{m_3}$ to the infinite line containing $\vec{m_{01}}$ (the vector from $\vec{m_0}$ to $\vec{m_1}$), let d be the distance from $\vec{m_0}$ to the intersection of the perpendicular and the line, and let τ be the rotation off the basis plane (the plane containing the three basis points).

Next, we derive an expression for the image position of the third model point, denoted $\vec{m_2}^{\pi}$, as a function

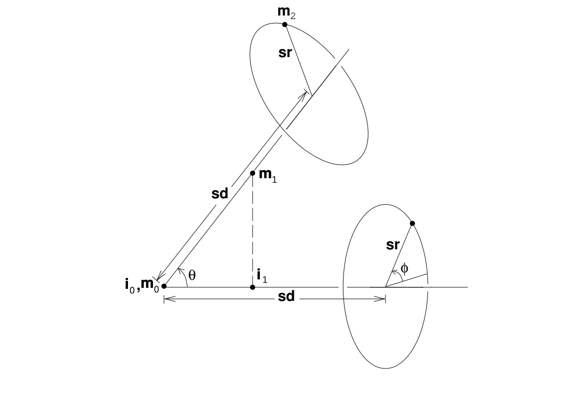

Figure 2: The position of $\vec{m_2}$ can be computed from its corresponding point on the circle with center on the x axis.

of two parameters (θ and ϕ) of a weak-perspective transformation that brings the model into the image. We may assume without loss of generality that initially $\vec{m_0} = \vec{i_0}, \vec{m_1} = \vec{i_1}$, and that $\vec{m_2}$ is in the $z = 0$ plane—in so doing, the model is scaled by $s_0 = \| \vec{i_{01}} \| / \| \vec{m_{01}} \|$. For the image coordinate system, we let the origin be $\vec{i_0}$, the z direction be orthogonal to the image plane, and the x axis be along $\vec{i_{01}}$. As in Fig. 2, next the model is scaled by s, then rotated by ϕ about the x axis, and then rotated by θ about the y axis (denoted by $\mathbf{R}_{\{\theta,y\}}$). This aligns the projections of the three model points with their corresponding image points. So $\vec{m_2} = \mathbf{R}_{\{\theta,y\}}(sd, sr\cos\phi, sr\sin\phi)$, which gives

$$\vec{m_2} = s \begin{bmatrix} d\cos\theta - r\sin\theta\sin\phi \\ r\cos\phi \\ d\sin\theta + r\cos\theta\sin\phi \end{bmatrix}$$

By our choice of image coordinate system, $\theta \in [0, \pi/2)$. Project orthographically:

$$\vec{m_2}^{\pi} = (x, y) = (sd\cos\theta, 0) + sr(-\sin\theta\sin\phi, \cos\phi),$$

As $\vec{m_1}$ rotates around the y axis, the scale factor is constrained to keep $\vec{m_1}$ projecting onto $\vec{i_1}$, so that $sd\cos\theta = s_0 d$ or $s = s_0/\cos\theta$. Consequently,

$$(x, y) = (s_0 d, 0) + s_0 r\left(-\tan\theta\sin\phi, \frac{\cos\phi}{\cos\theta}\right), \quad (1)$$

which gives the image position of $\vec{m_2}$ as a function of the "out-of-plane" rotation angles θ and ϕ. As ϕ changes, (x, y) traces out an ellipse in the image. As θ changes, (x, y) traces out a hyperbola [12].

If $\vec{i_2}$ alone moves in the plane, the out-of-plane rotations, θ and ϕ, and the scale needed to align the three points will change. If the changes in θ and ϕ are sufficiently small, then the changes in x and y as a function of θ and ϕ will be given by their first derivatives and will be independent. From Eq. 1,

$$\frac{\partial x}{\partial \theta} = -s_0 r \frac{1}{\cos^2\theta}\sin\phi, \qquad \frac{\partial y}{\partial \theta} = s_0 r \frac{\sin\theta}{\cos^2\theta}\cos\phi$$
$$\frac{\partial x}{\partial \phi} = -s_0 r \frac{\sin\theta}{\cos\theta}\cos\phi, \qquad \frac{\partial y}{\partial \phi} = -s_0 r \frac{1}{\cos\theta}\sin\phi$$

As θ changes, s changes implicitly to keep $\vec{m_1}$ projecting onto $\vec{i_1}$. $\vec{t_\theta} = (\partial x/\partial\theta,\ \partial y/\partial\theta)$ and $\vec{t_\phi} = (\partial x/\partial\phi,\ \partial y/\partial\phi)$ are the tangents at the point (x, y) to the image curves traced out by changing θ and ϕ.

$$\|\ \vec{t_\theta}\ \| = \frac{s_0 r}{\cos^2\theta}\sqrt{\sin^2\phi + \sin^2\theta\cos^2\phi} \qquad (2)$$

$$\|\ \vec{t_\phi}\ \| = \frac{s_0 r}{\cos\theta}\sqrt{\sin^2\phi + \sin^2\theta\cos^2\phi} \qquad (3)$$

For a small change in θ, let α_θ represent the direction in which the image point $\vec{i_2}$ moves, measured counter-clockwise from the x axis, and similarly for ϕ and α_ϕ. Then $\alpha_\theta = \tan^{-1}(\partial y/\partial\theta, \partial x/\partial\theta)$ and $\alpha_\phi = \tan^{-1}(\partial y/\partial\phi, \partial x/\partial\phi)$, or

$$\alpha_\theta = \tan^{-1}(\sin\theta\cos\phi, -\sin\phi) \qquad (4)$$
$$\alpha_\phi = \tan^{-1}(-\sin\phi, -\sin\theta\cos\phi) \qquad (5)$$

The dot product of the arguments to the inverse tangent is 0, and so the tangent vectors $\vec{t_\theta}$ and $\vec{t_\phi}$ are perpendicular. The normalized cross product of the arguments is 1, and so $\sin(\alpha_\phi - \alpha_\theta) = 1$, or $\alpha_\phi = \alpha_\theta + 90°$.

Eqs. 2 and 3 give the magnitude of the changes in $\vec{m_2}^\pi$ for small changes in θ and ϕ. The only differences for $\vec{m_3}^\pi$ are in the relative positions of the model points (Fig. 1). For $\vec{m_2}$, the position is given by $(r, d, 0)$. For $\vec{m_3}$, let these values be (r', d', τ'). In the expressions, we change $r \to r'$, $d \to d'$, $\phi \to \phi + \tau'$, $\vec{t_\phi} \to \vec{t'_\phi}$, and $\vec{t_\theta} \to \vec{t'_\theta}$. From Eqs. 2 and 3, $\|\ \vec{t'_\theta}\ \| \,/\, \|\ \vec{t_\theta}\ \| = \|\ \vec{t'_\phi}\ \| \,/\, \|\ \vec{t_\phi}\ \| = S$, or

$$S = \frac{r'}{r}\sqrt{\frac{\sin^2(\phi+\tau') + \sin^2\theta\cos^2(\phi+\tau')}{\sin^2\phi + \sin^2\theta\cos^2\phi}} \qquad (6)$$

Therefore for any change in $\vec{m_2}^\pi$ by some amount, there is a change in $\vec{m_3}^\pi$ by that amount times S.

Using the normalized cross product again, we obtain that $\sin(\alpha'_\phi - \alpha'_\theta) = 1$, and so

$$\alpha'_\phi - \alpha'_\theta = \alpha_\phi - \alpha_\theta = 90°. \qquad (7)$$

This tells us that to a first approximation, as θ changes (and the scale s changes to keep $\vec{m_1}^\pi$ fixed), $\vec{m_2}^\pi$ moves in some direction, while $\vec{m_3}^\pi$ moves in a different direction by a different amount. And then as ϕ changes the two points move at right angles to their previous directions and with the same relative speeds. In addition, with $\vec{m_0}^\pi$ and $\vec{m_1}^\pi$ fixed, θ and ϕ are the only parameters of $\vec{m_2}^\pi$ and $\vec{m_3}^\pi$ that change. And, any movement of $\vec{m_2}^\pi$ in the image plane can be viewed as the sum of the effects of changes in θ and ϕ, since we are making a first order approximation. This implies that any change in $\vec{m_2}^\pi$ produces a change in $\vec{m_3}^\pi$ that is rotated and scaled by a fixed amount. Thus any error region about the nominal position of $\vec{m_2}^\pi$ results in a mathematically similar error region about the nominal position of $\vec{m_3}^\pi$.

We can explicitly write the relationship between the errors in $\vec{m_2}^\pi$ and $\vec{m_3}^\pi$ using a 2×2 scaled rotation matrix $\mathbf{A}$: $\mathbf{A}$ must satisfy

$$\vec{t'_\theta} = \mathbf{A}\vec{t_\theta} \quad \text{and} \quad \vec{t'_\phi} = \mathbf{A}\vec{t_\phi}. \qquad (8)$$

from which we can derive (see [12]):

$$a_{11} = k(\cos\tau' - \cos^2\theta\cos\phi\cos(\phi+\tau')) \qquad (9)$$
$$a_{12} = -k\sin\tau'\sin\theta, \qquad (10)$$
$$k = \left(\frac{r'}{r}\right)\frac{1}{1-\cos^2\theta\cos^2\phi} \qquad (11)$$

Note that the constant scale S from Eq. 6 must equal $\sqrt{a_{11}^2 + a_{12}^2}$.

We can use Eqs. 9-11 to obtain a formula for the error in $\vec{m_3}^\pi$ as a function of the error in $\vec{i_0}$ or $\vec{i_1}$, in the same way. Under a first-order approximation, these errors affect the error in $\vec{m_3}^\pi$ independently and the total error in $\vec{m_3}^\pi$ is the sum of the individual errors. Let $\mathbf{A}, \mathbf{B}, \mathbf{C}$ be the scaled rotation matrices relating $\vec{i_0}, \vec{i_1}, \vec{i_2}$, respectively, to $\vec{m_3}^\pi$. Also let $\vec{e_0}$, $\vec{e_1}$, and $\vec{e_2}$ be the errors in the basis points. Then the error in $\vec{m_3}^\pi$ is given by the following linear relationship:

$$\vec{e_3} \approx \mathbf{A}\vec{e_0} + \mathbf{B}\vec{e_1} + \mathbf{C}\vec{e_2} \qquad (12)$$

The two weak-perspective pose solutions, $\pm(\theta, \phi)$, lead to two possibilities for each of $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, and for the error in the fourth point $\vec{e_3}$.

Suppose now that the error in each image point is bounded by some amount ϵ. For each image point separately, its ϵ-circle propagates to a circle around $\vec{m_3}^\pi$ that is scaled by S (Eq. 6), which gives a radius of $S\epsilon$. The error in each image point affects the fourth point independently, and so the uncertainty region around the fourth point is a circle centered at $\vec{m_3}^\pi$, and the three radii simply sum together (where we must be careful to use the radii from the same weak-perspective solution). In Eq. 12, let $S_0 = \sqrt{a_{11}^2 + a_{12}^2}$, $S_1 = \sqrt{b_{11}^2 + b_{12}^2}$ and $S_2 = \sqrt{c_{11}^2 + c_{12}^2}$. Then the radius of the error circle for $\vec{m_3}^\pi$ is

$$R = S_0\epsilon_0 + S_1\epsilon_1 + S_2\epsilon_2 + \epsilon_3 \qquad (13)$$

Error	2	4	6	8	10	12
Circles	86.7	94.2	97.1	98.2	98.8	99.0

Table 1: The percent of uncertainty circles for which the relative error between the circles predicted by our formula and by the method in [1] method is less than a given percent.

where ϵ_3 is added to account for error in sensing $\vec{i}_3$ (an image point that corresponds to $\vec{m_3}^\pi$).

When the error in the image points is normally distributed, the linear relationship allows us to determine the propagated uncertainty distribution about any fourth point. If the Gaussian error in the ith image point has standard deviation σ_i, then the uncertainty distribution about $\vec{m_3}^\pi$ is normally distributed with standard deviation

$$\sigma = \sqrt{S_0^2\sigma_0^2 + S_1^2\sigma_1^2 + S_2^2\sigma_2^2 + \sigma_3^2}. \tag{14}$$

(see [12]). This result could be used to extend [16] to the full 3D case.

3 Testing the Linear Approximation

We have run two experiments to test the accuracy of the results in Sec. 2. The first experiment compares our results to those of [1]. [1] showed experimentally that the uncertainty region for a fourth model point is closely approximated by a circle centered at the nominal point. To see how our uncertainty circles compare, we ran a series of trials like those in [1]: We tested 1,163 uncertainty circles, over which the average relative error was 1.45%. Table 1 shows the percent of uncertainty circles for which relative error was less than some threshold. The results show that our circles reasonably approximate the maximum-radius circles from [1], which showed that the maximum-radius circles reasonably approximate the true regions. See [12] for further details.

Sec. 2 showed how we can approximate the effects of error in the basis points using three similarity transforms (Eq. 12). The next experiment estimates how well this approximation works when the error in the image points is distributed uniformly, or according to a Gaussian. We ran a series of trials where at each trial we generated a random model and a random image triple. First assuming uniform error, we perturbed each image point within a circle of radius ϵ. Using the perturbed image points, we computed the distances between where the similarity transforms predict the fourth model point will appear and its actual location. For each unmatched model point, there were two such distances corresponding to the two weak-perspective solutions. For each trial, there were 7 unmatched model points giving 14 distances. Over 10,000 trials, 140,000 unmatched model points were tested.

Distance	1	2	3	4	5
Uniform	98.25	99.60	99.79	99.87	99.90
Gaussian	98.53	99.64	99.82	99.88	99.91

Table 2: For uniform and Gaussian error, the percent of unmatched model points for which the distance between the similarity point and the true point is less than some tolerated number of pixels.

For uniform error, the average distance between the similarity point (the point predicted by the similarity transforms) and the true point was .06 pixels. Table 2 gives the percent of unmatched points for which the distance was less than various amounts of tolerated error. The results show that the similarity transforms very accurately predict the movement of the fourth point, with 98% probability of being in error by less than one pixel.

For Gaussian error, we ran the same experiment but instead sampled a 2D Gaussian distribution, using a standard deviation of 2.5 pixels. We still used a bound of $\epsilon = 5$ pixels on the allowed error in the image points. The results were similar (Table 2).

In conclusion, the two experiments on random model and image points indicate that the linear approximation is reasonably accurate for up to five pixels of error. Fig. 3 shows rectangular uncertainty regions that we computed using a real model and image of a telephone. By hand, we measured corners of the telephone to obtain a model and selected corresponding corners in the image. The three smaller circles are the error bounds for the matched image points. The remaining circles are the correct search regions for finding additional matches.

4 nth-Point Uncertainty Region

It has been shown [8] that recognition algorithms that use a small number of point matches to determine pose are sensitive to false positive identifications of objects, because these poses are not sufficiently stable, and lead to large uncertainty regions. It is therefore important to use poses based on more information to derive smaller uncertainty regions. Assuming a bounded error model, this section shows how to compute the uncertainty region of an $n+1$'st model point given n matched model and image points, for any value of n. Our linearized model of weak-perspective projec-

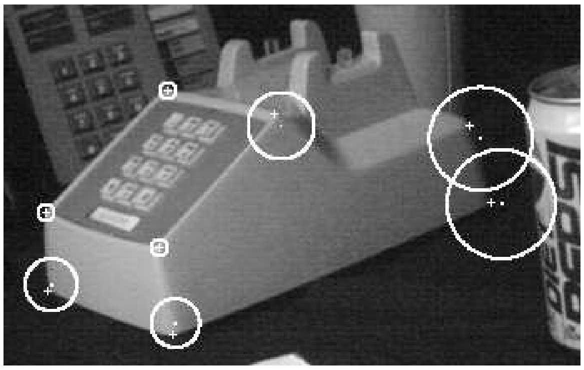

Figure 3: Circular uncertainty regions computed for points at the corners of a telephone. Small crosses show the actual locations of the image points, which were selected by hand, but are a bit noisy. Small dots show the projected locations of the model points in the determined pose.

tion allows us to derive linear constraints on the set of feasible error vectors consistent with a match between model and image points. Having expressed our knowledge about pose using linear constraints, we use linear programming to optimize a set of objective functions whose solution tightly bounds the uncertainty region. In general, this technique applies to any linear projection model, even though here we use a linearized, weak-perspective model.

To demonstrate the idea, we suppose that the error in each image point is bounded by a square of width 2ϵ, but we emphasize that the same reasoning applies to any convex, polygonal error bound, so that we may approximate a circle, or any other convex error bound, as closely as we wish. A match between image and model points can be consistent, then, only if there exists a pose that brings all associated coordinates to within ϵ pixels of each other. Let $\vec{m_i}^\pi = (x_i^\pi, y_i^\pi)$ be the projection of the i'th matched model point, in some nominal pose. Let $\vec{i_i} = (x_i, y_i)$ be the nominal location of the corresponding image point. Also, define $\vec{e_i} = (e_{\{i,x\}}, e_{\{i,y\}})$ so that $\vec{m_i}^\pi + \vec{e_i}$ is the true, error-free position of each model point in the image. Then we model error by assuming:

$$
\begin{aligned}
x_i - \epsilon \le x_i^\pi + e_{\{i,x\}} \le x_i + \epsilon \\
y_i - \epsilon \le y_i^\pi + e_{\{i,y\}} \le y_i + \epsilon
\end{aligned}
$$

We choose the nominal pose of the object so that for the first three matched points $\vec{i_i} = \vec{m_i}^\pi$. Consequently, for $i \in [0, 2]$ we have the following constraints on the error in the first three points:

$$
-\epsilon \;\le\; e_{\{i,x\}} \le \epsilon, \qquad -\epsilon \le e_{\{i,y\}} \le \epsilon.
$$

For $i \in [3, n]$, Eq. 12 leads to additional constraints:

$$
\begin{aligned}
x_i - \epsilon \;\le\;& x_i^\pi + a_{\{11,i\}} e_{\{0,x\}} + a_{\{12,i\}} e_{\{0,y\}} \\
& + b_{\{11,i\}} e_{\{1,x\}} + b_{\{12,i\}} e_{\{1,y\}} \\
& + c_{\{11,i\}} e_{\{2,x\}} + c_{\{12,i\}} e_{\{2,y\}} \\
\le\;& x_i + \epsilon, \\
y_i - \epsilon \;\le\;& y_i^\pi - a_{\{12,i\}} e_{\{0,x\}} + a_{\{11,i\}} e_{\{0,y\}} \\
& - b_{\{12,i\}} e_{\{1,x\}} + b_{\{11,i\}} e_{\{1,y\}} \\
& - c_{\{12,i\}} e_{\{2,x\}} + c_{\{11,i\}} e_{\{2,y\}} \\
\le\;& y_i + \epsilon.
\end{aligned}
$$

We have formulated our knowledge of model pose, based on a match between n image and model points, in terms of linear constraints on the components of the three error vectors $\vec{e_0}$, $\vec{e_1}$, $\vec{e_2}$. This formulation is similar to that of [2]. Also relevant is [4]'s work on polynomial time matching with bounded error. In fact, our results can be used to extend [2], [4], [11] and [3] to scaled orthographic projection (see [12]).

We may now use linear programming to maximize any linear objective function, subject to these constraints. In particular, we can formulate linear objective functions that express the position that an additional model point may project to in the image, for some direction, and extremize these functions. For example, if we maximize the linear objective function:

$$
\begin{aligned}
& x_{n+1}^\pi + e_{\{n+1,x\}} = x_{n+1}^\pi + a_{\{11,n+1\}} e_{\{0,x\}} + \\
& \quad a_{\{12,n+1\}} e_{\{0,y\}} + b_{\{11,n+1\}} e_{\{1,x\}} + \\
& \quad b_{\{12,n+1\}} e_{\{1,y\}} + c_{\{11,n+1\}} e_{\{2,x\}} + c_{\{12,n+1\}} e_{\{2,y\}}
\end{aligned}
$$

we will find the maximum x value that the projection of the $n+1$'st model point can have. By maximizing the negation of this expression, and similar expressions for the y values, we may put a rectangle about the possible locations of the $n+1$'st point. In general, we can place a convex polygon of any shape about the possible locations of the $n+1$'st point [12].

This algorithm could be quite useful to alignment-type approaches to recognition [7, 14, 10], or in methods that match image to model features by indexing and then verify these matches [17]. Recognition systems commonly use more complex features such as line segments for verification, instead of points. It would be straightforward, however, to use our results to bound the uncertainty regions of line segments by finding the uncertainty region of their endpoints (as in [1]). A complete recognition system using this result might work in the following steps: 1) match k image and model points, using a search or indexing method. Assume error in each image point is bounded by some m-sided polygon. 2) Use these matches to derive km linear constraints on the possible errors in the first three matched points. 3) For each unmatched model point, run l linear programs to compute an l-gon bounding this point's possible image location. Look there for a matching image point. Additionally, our results allow us to run experiments measuring the extent to which additional point matches decrease our uncertainty about the location of unmatched points.

5 Experiments

To demonstrate the value of narrowing the feasible region in which model points may appear by using additional matches, we have implemented a test system. In this system, we match some image and model points. We then examine the regions in which additional points might appear. We can see how much smaller these feasible regions get as we derive additional constraints using more matches.

We first describe the results of this system on synthetic data. We use the following experimental conditions. First, we generate random sets of seven model points inside a cube. We form an image by projecting these points orthographically, and scaling so that the cube projects to a 1000×1000 square image. We then add error so that each sensed point shows up inside a circle of radius five pixels centered at the projected position of the point. For error, a uniform distribution is used.

We then match the first three noisy image points and model points, and use this match to generate a noisy pose of the model. Of the two possible model poses that can be derived using a match of three points, we automatically select the one that is closer to the correct pose. In a real system, of course, we would have to explore both possibilities, and see which leads to more confirming evidence.

Given these linear transformations, we allow for a square error bound of width 10 pixels around each error circle. As described above, we then compute a rectangular bound on the location of each additional model point in the image using linear programming. Next we check to see which image points actually appear within the predicted rectangular boundary. Since we have perturbed the image points within their error circles, any time we fail to find a model's image point in the predicted rectangle, this mistake must be due to limitations in our linear approximation. When we do find an image point, we record the size of the rectangle in which we looked.

We then augment our hypothesis by matching additional image and model points, and, using the additional constraints, we further narrow down the locations of the remaining model points in the image. We repeat this experiment $2,500$ times, continuing to perturb each image point by up to five pixels, but allowing for varying levels of error in our predictions. We can use these results to see how many mistakes of the system could be eliminated by overestimating the expected error, and how much of a price we would pay for this by producing larger rectangles.

Error Allowed for	Num. Pts. Matched	Rectangle Avg. Area	Percent Correct
5.0	3	16,700	98.9
	4	1,460	97.6
	5	823	96.2
	6	599	94.6
5.25	3	18,300	99.0
	4	1,600	97.8
	5	898	96.7
	6	655	95.5
5.5	3	10,800	98.9
	4	1,750	98.0
	5	969	97.0
	6	730	96.4
6.0	3	12,700	99.2
	4	2,200	98.8
	5	1,370	98.4
	6	948	97.9

Table 3: This shows the size of error regions computed as more points are matched, and the frequency with which noisy model points fail to appear in these error regions. The first column gives half the width of the square error bound that we allow for in each image point. The second column gives the number of matches used in computing the error region for an additional point. The third column gives the average size of this error region, and the fourth column gives the percentage of times that a model point's image shows up in this predicted error region.

Table 3 lists the results. There are several conclusions we may draw. First, we see that few overall mistakes are made. Second, we can see that additional matches do provide considerable extra constraint in determining the location of unmatched points. These results also help us to in general assess the stability of poses generated by a small number of feature matches. We can see that if we use three points to compute a pose, small changes in these points can result in large changes in the locations of additional points.

As before, we have run this system on a real model and image to further illustrate its performance. Fig. 5 shows the resulting rectangular error regions for $\epsilon = 5.25$. The figure demonstrates how the un-

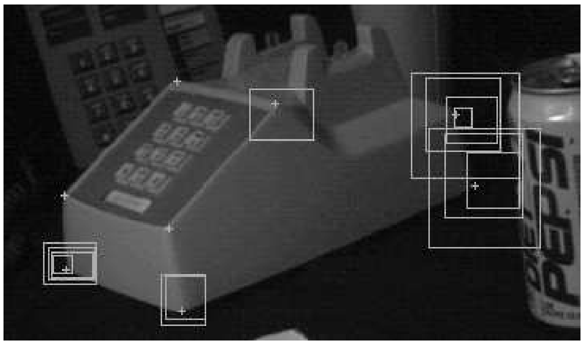

Figure 4: Rectangular uncertainty regions from matching more points. The outer rectangles are computed after matching only three points. As we match each additional point, we stop computing rectangles for that point, and compute smaller rectangles for the remaining points.

certainty regions shrink as we match more points, while still containing the true image point. These experiments demonstrate that we can use additional matched points to significantly reduce our uncertainty as to the location of still unmatched model points, with little loss in accuracy.

6 Conclusions

We have shown how to compute the propagated uncertainty regions for the rigid projection of a 3D model into a 2D image. When three points are matched, the uncertainty regions, under Gaussian or bounded error, are described by a simple analytic expression. When more points are matched, for bounded error, we present a simple and efficient algorithm for computing the uncertainty regions. This work will allow 3D recognition systems to accurately take account of the effects of sensing error in a verification process that finds supporting evidence to confirm a hypothetical set of matches. Our experiments also demonstrate that we can obtain accurate uncertainty regions that are significantly smaller when we have matched more than three points.

We have also shown that a linear approximation to scaled orthographic projection is accurate when reasonable amounts of sensing error have occurred. Consequently we can extend several approaches to handling error in recognition systems that were previously restricted to domains having linear projection models.

References

[1] Alter, T. D., & W. E. L. Grimson, "Fast and Robust 3D Recognition by Alignment," *Proc. 4th Inter. Conf. Computer Vision*, May 1993.

[2] Baird, H. S., *Model-Based Image Matching Using Location*, Cambridge, MA: MIT Press, 1985.

[3] Breuel, T., "Model Based Recognition using Pruned Correspondence Search," *IEEE Conf. on Comp. Vis. Pat. Rec.*, pp. 257-268, 1991.

[4] Cass, T., "Polynomial Time Object Recognition in the Presence of Clutter, Occlusion and Uncertainty," *2nd Eur. Conf. on Comp. Vis.*, pp. 834-842, 1992.

[5] Costa, M., R. M. Haralick, & L. G. Shapiro, "Optimal Affine-Invariant Point Matching," *Proc. 6th Israeli Conf. on AI*, pp. 35-61, 1990.

[6] Ellis, R. E., "Uncertainty Estimates for Polyhedral Object Recognition," *Proc. IEEE Conf. Rob. Aut.*, pp. 348-353, 1989.

[7] Fischler, M., & R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Analysis and Automated Cartography," *Comm. of the Assoc. of Comp. Mach.*, **24**(6):381-395, 1981.

[8] Grimson, W. E. L., D. P. Huttenlocher, & T. D. Alter, "Recognizing 3D Objects from 2D Images: An Error Analysis," *IEEE Conf. Comp. Vis. Pat. Rec.*, 1992.

[9] Grimson, W. E. L., D. P. Huttenlocher, & D. W. Jacobs, "A Study of Affine Matching with Bounded Sensor Error," *Proc. 2nd European Conf. Computer Vision*, pp. 291-306, May 1992.

[10] Huttenlocher, D., & S. Ullman, "Recognizing Solid Objects by Alignment with an Image," *Inter. Journal of Computer Vision*, **5**(2):195-212, 1990.

[11] Jacobs, D. W., "Optimal Matching of Planar Models in 3D Scenes," *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 269-274, 1991.

[12] Jacobs, D. W., & T. D. Alter, AI Memo 1476, 1994.

[13] Lamdan, Y., & H. J. Wolfson, "On the Error Analysis of 'Geometric Hashing'," *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 22-27, 1991.

[14] Lowe, D., *Perceptual Organization and Visual Recognition*, The Netherlands: Kluwer Academic Publishers, 1985.

[15] Rigoutsos, I., & R. Hummel, "Robust Similarity Invariant Matching in the Presence of Noise," *8th Israeli Conf. Artif. Intell. Comp. Vis.*, Tel Aviv, 1991.

[16] Sarachik, K. B., & W. E. L. Grimson, "Gaussian Error Models for Object Recognition," *Proc. IEEE Conf. Comp. Vis. Pat. Rec.*, pp. 400-406, June 1993.

[17] Thompson, D. W., & J. L. Mundy, "Three-Dimensional Model Matching from an Unconstrained Viewpoint," *Proc. IEEE Conf. Rob. Aut.*, 1987.