

Figure 5: A plot where six points are given and the position of the seventh point is estimated. The graph shows the percentage of times that the seventh point's image shows up in the predicted error region and the average area of the uncertainty region.

- [5] R.V. Hogg and A.T. Craig. Introduction to Mathematical Statistics. Macmillan Publishing, New York, 1978.
- [6] D. Huttenlocher and S. Ullman. Recognizing 3D solid objects by alignment with an image. Int. J. of Comp. Vision, 5(2):195-212, 1990.
- [7] D.W. Jacobs. Optimal matching of planar models in 3d scenes. In Proc. IEEE Conf. Comp. Vision Patt. Recog., pages 269-274, 91.
- [8] D. Kahaner, C. Moler, and S. Nash. Numerical Methods and Software. Prentice Hall, Englewood Cliffs, NJ, 1989.
- [9] I. Shimshoni and J. Ponce. Probabilistic 3D object recognition. In Proc. Int. Conf. Comp. Vision, pages 488-493, Boston, Massachusetts, June 1995.



Figure 6: For a given probability for the point in the image to be in the uncertainty region, the average area of the uncertainty region is shown in logarithmic scale for different sizes of sets of matched points.



Figure 7: Uncertainty regions shrink as more points are added to the match. The larger circles are computed when only three points are matched. As more points are added the uncertainty regions shrink but still contain the measured point.

never requires an average area of 1750 pixels, because it reaches 100% success rate for an average area of 880 pixels.

In Figure 6 we plot the size of the uncertainty region as a function of the size of the match for several values of the success rate. This graph which is plotted on a logarithmic scale demonstrates how dramatically the size of the uncertainty region shrinks as the size of the match increases.

Finally we have tested our system on an image of a real object whose corners we have measured by hand. In Figure 7, the uncertainty regions are overlayed over the image. We assume that  $\epsilon = 7.7$  pixels and chose  $\lambda = 3.0$ . At first we computed the uncertainty regions of all the points using a match of size three. We then added additional points to our match and computed the uncertainty regions for the remaining points. We continue this process until all the points measured in the image are part of the match. The figure demonstrates how the uncertainty decreases significantly when the size of the match increases.



Figure 3: A plot where four points are given and the position of the fifth point is estimated. The graph shows the percentage of times that the fifth point's image shows up in the predicted error region and the average area of the uncertainty region.

# 5 Discussion and Future Work

We have presented an extremely efficient method for estimating the uncertainty region of the projection of a model point in the image when a match of three other model points is given. We then incorporated our method into a general statistical framework to deal with the case when more points are matched. Besides being very efficient our method also estimates



Figure 4: A plot where five points are given and the position of the sixth point is estimated. The graph shows the percentage of times that the sixth point's image shows up in the predicted error region and the average area of the uncertainty region.

the uncertainty regions better than previously published methods by finding the optimal linear combination of three point estimates which produces a very good estimate for the position of the projected model point, and a very small variance.

Future work will be dedicated to using this method in recognition systems. The false negative and false positive rates of our method can be chosen by the user of our method. Therefore, further analysis has to be performed on what combination of them will make the recognition system most efficient.

#### References

- T. D. Alter. 3D pose from 3 points using weakperspective. *IEEE Trans. Patt. Anal. Mach. Intell.*, 16(8):802-808, August 1994.
- [2] T. D. Alter and D. W. Jacobs. Error propagation in full 3D-from-2D object recognition. In Proc. IEEE Conf. Comp. Vision Patt. Recog., pages 892-898, Seattle, Washington, June 1994.
- [3] T.D. Alter and W.E.L. Grimson. Fast and robust 3d recognition by alignment. In Proc. Int. Conf. Comp. Vision, pages 113-120, 93.
- [4] W. E. L. Grimson, D. P. Huttenlocher, and T. D. Alter. Recognizing 3D objects from 2D images; an error analysis. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, pages 316–321, Champaign, Illinois, June 1992.

with the smallest possible variance (i.e. smallest uncertainty region).

A theorem in statistics [5] states the following. Let  $X_1, \dots, X_m$  denote random variables that have means  $\mu_1, \dots, \mu_m$  and variances  $\sigma_1^2, \dots, \sigma_m^2$ . Let  $\rho_{ij}$ ,  $i \neq j$ , denote the correlation coefficient of  $X_i$  and  $X_j$ and let  $K_1, \dots, K_m$  denote real constants. The mean and the variance of the linear function

$$Y = \sum_{1}^{m} K_i X_i$$

are respectively

$$\mu_Y = \sum_1^m K_i \mu_i$$

and

$$\sigma_Y^2 = \sum_{1}^{m} K_i^2 \sigma_i^2 + 2 \sum_{i < j} K_i K_j \rho_{ij} \sigma_i \sigma_j \qquad (11)$$

In our case all the means are equal to each other and for Y to have the same mean  $\sum_{i=1}^{m} K_i$  must equal to one. When  $X_1, \dots, X_m$  are mutually stochastically independent the minimal variance for Y is achieved when

$$K_i = \frac{1}{\sigma_i^2} \frac{1}{\sum_{1}^m \frac{1}{\sigma_i^2}}$$

However, as the match triples are usually not disjoint the  $X_i$ 's are not independent, therefore these values for the  $k_i$ 's will not be optimal.

In order to analyze the problem of finding the optimal weights K consider the following two observations. First, each of the  $\binom{n}{3}$  estimation regions is approximately circular. Therefore a linear combination of those estimation regions would also be a circular. We can therefore limit ourselves to finding weights Kthat minimize the extent of the x coordinate of the uncertainty region. And second, all pairs of the 2ncoordinates  $\xi$  of the n matched image points have a correlation coefficient zero (including the x and y coordinate of the same point), and have a variance  $\epsilon^2/4$ .

Using our linear approximation of the x coordinate of the image position of the n + 1'th point developed in Section 3 we can estimate the variance of the *i*'th estimate using (11). We shall denote by  $A_i$  a vector of coefficients of the coordinates  $\xi$ , all but six of which are zero, such that if  $X_i$  is the *i*'th estimate

$$X_i(\xi + \Delta \xi) \approx X_i(\xi) + A_i \Delta \xi$$

Using (11) we get

$$\sigma_{X_i}^2 \approx \frac{\epsilon^2}{4} |A_i|^2$$

Performing the same analysis for the linear combination of the  $X_i$ 's yields

$$Y = \sum_{1}^{m} K_{i} X_{i} = \sum_{1}^{m} K_{i} X_{i}(\xi) + \sum_{1}^{m} K_{i} A_{i} \Delta \xi$$

Writing this in matrix form where the matrix  $A = (A_1 \cdots A_m)$  and  $\mu_x = X_i(\xi)$  yields

$$Y = \mu_x + AK\Delta\xi$$

where AK is the vector of coefficients of the coordinates  $\xi$ . Therefore using (11) again yields

$$\sigma_Y^2 = \frac{\epsilon^2}{4} |AK|^2 \tag{12}$$

Thus we have to find a K that minimizes (12) under the constraint  $\sum_{i=1}^{m} K_i = 1$ . Substituting  $K_1 = 1 - \sum_{i=1}^{m} K_i$  into (12) yields

$$\sigma_Y^2 = \frac{\epsilon^2}{4} \left| \left( (A_2 - A_1) \cdots (A_m - A_1) \right) \begin{pmatrix} K_2 \\ \vdots \\ K_m \end{pmatrix} + A_1 \right|^2$$
(13)

(13) is the formulation of a standard linear least squares problem which is solved exactly using standard numerical methods [8]. The only problem with this solution is that the  $X_i$ 's are not linear functions of the coefficients  $\xi$  and thus our model for the variance of the linear combination of the  $X_i$ 's is not exact either. The major problem is that the  $K_i$ 's are not bounded. They could have arbitrarily large negative or positive values which would magnify the small error in our linear model of the  $X_i$ 's and increase  $\sigma_Y^2$ . To deal with this problem we add to each coefficient vector  $A_i$  a small coefficient for an additional dummy random variable which appears only in that  $X_i$ . This variable acts as a penalty against large values for the  $K_i$ 's.

The value obtained for  $\sigma_Y$  and the vector K are returned by the algorithm. The user chooses the value of the parameter  $\lambda$  which is the radius of the uncertainty region in standard deviations.

The results for match sizes four, five, and six are traced in Figures 3,4 and 5 respectively for several values of  $\lambda$ . We plot the results presented in [2] and our results. To demonstrate the quality of our results consider the following example. In order for 98% of the uncertainty regions to contain the image of the model point for a match of size four, the method presented in [2] requires that the average area of the uncertainty region be 1750 pixels, where our method requires an average area of only 480 pixels. Our method

Method	Cost	Better
[3]	512 Pose Estimations	
[2]	Linear Approximation	N/A
Basic Method	2 Pose Estimations	70.6
Derivative Method	8 Pose Estimations	84.7
Combined Method	9 Pose Estimations	83.9
Minimization	50 Pose Estimations	85.9
Rand. Samp. 1	512 Pose Estimations	71.0
Rand. Samp. 2	175 Pose Estimations	49.6

Table 1: The cost of estimating uncertainty regions, and the percent of uncertainty circles for which the method gives a better estimate than [3].

$\mathbf{M}$ ethod	2	4	6	8	10
Basic Method	85.7	88.7	91.0	92.1	93.4
Derivative Method	95.0	96.7	97.5	97.8	98.3
Combined Method	95.1	96.7	97.5	97.8	98.3
Minimization	97.1	98.5	99.1	99.3	99.6
Rand. Samp. 1	98.4	99.7	99.8	99.8	99.9
Rand. Samp. 2	87.9	96.5	98.9	99.4	99.6

Table 2: The percent of uncertainty circles for which the methods underestimate the area of the uncertainty region relative to the method in [3] by less than a given percent.

image point is within our uncertainty region. When the image point is found, we record the size of region in which we have been looking.

The performance of recognition algorithms which would use our algorithm depend on the false positive and false negative rates of our method. We therefore let the designer of the recognition algorithm have a choice between several values of false positive and false negative rates to choose from. We therefore multiply the results of our estimation algorithms by a user supplied parameter  $\lambda$  and use the result as the estimate. The larger the value of  $\lambda$  the higher the false positive rate and the lower the false negative rate. The designers of recognition algorithms will be able to choose the value of  $\lambda$  which produces the most efficient algorithm.

We plotted a graph for the success rate and average uncertainty area for the method presented in [2], and for the first four methods described above in Figure 2. To analyze this graph we have to check the "price" that has to be paid in the average area of the uncertainty region for a given success rate and vice versa. From this analysis we can see that the minimization algorithm performs the best for very low false negative rates. For all other values of false negative rates, the combined method performs the best with the pure derivative method yielding similar results, the basic method is next, and finally the method presented in [2].

One of the strengths of the method presented in [2] is that it has been extended to deal with sets of matches of any size using linear programming. In the next section we will employ statistical analysis to extend our method to deal also with larger sets of matches.



Figure 2: A plot where three points are given and the position of the fourth point is estimated. The graph shows the percentage of times that the fourth point's image shows up in the predicted error region and the average area of the uncertainty region.

#### 4 nth-Point Uncertainty Region

It has been shown [2, 4] that recognition algorithms which use a small number of features to estimate the pose of the object encounter large numbers of false positive matches due to the uncertainty in the pose which leads to large uncertainty regions in which the wrong image features may lie. Therefore it is important to derive methods to compute more exact poses based on more information which yield smaller uncertainty regions.

Given a match of size n there are  $\binom{n}{3}$  different matches of size three. For each one of them we can use the combined method to estimate the position of the n + 1'th point and estimate the radius of the uncertainty region. Putting this into a statistical context, we have  $\binom{n}{3}$  random variables whose mean is the position of the n + 1'th point and whose variance is approximately proportional to the area of the uncertainty region for which we have an estimate. Our goal is to combine all these estimates into a single estimate above are all rotated by any angle  $\theta$  the displacement stays the same, yielding a circular uncertainty region.

The final stage in our estimate computation is to reevaluate  $H_1$  and  $H_2$  using as the image positions  $\mathbf{i}_0 + \epsilon_0$ ,  $\mathbf{i}_1 + \epsilon_1$ , and  $\mathbf{i}_2 + \epsilon_2$ , compute the perturbed and unperturbed position of  $\mathbf{m}_3$  in the image and the distance between them. This distance is used as our estimate for the radius of the uncertainty region.

Our estimation method underestimates the real radius of the uncertainty region. However it is extremely fast, requiring only two pose estimations which are done using Alter's method which is also very efficient.

In order to get a better estimate we have to analyze the effects of the changes in  $\mathbf{i}_0, \mathbf{i}_1$  and  $\mathbf{i}_2$  on  $H_1$  and  $H_2$ . Let  $\xi$  denote the vector of the six coordinates of  $\mathbf{i}_0, \mathbf{i}_1$  and  $\mathbf{i}_2$ . We can evaluate the derivatives

$$\frac{\partial H_i}{\partial \xi_j}$$
  $i = 1 \cdots 2$   $j = 1 \cdots 6$ 

by computing directly the derivatives of the equations in Section 2, or by estimating them using finite differences at a cost of six additional pose estimations. We use these derivatives for the following first order approximation of  $H_1$  and  $H_2$ 

$$H_i(\xi + \Delta \xi) = H_i(\xi) + \sum_{j=1}^{6} \Delta \xi_j \frac{\partial H_i}{\partial \xi_j} + o(\epsilon^2)$$

Substituting these equations into (8) we get a more exact first order approximation for the x coordinate of  $\mathbf{m}_3$ . To the coefficients of the perturbations computed above we add

$$\gamma(y_1 - y_0)\frac{\partial H_2}{\partial \xi_i} - \gamma(y_2 - y_0)\frac{\partial H_1}{\partial \xi_i}.$$

We use the method described above to compute the  $\epsilon_i$ 's from the coefficients and compute the estimate using the perturbed points.

#### 3.1 Experimental Results

In order to test our method, we present here the results of running experiments on several variants of our method and comparing them to previously published methods.

In [3], the three uncertainty circles of radius  $\epsilon$  are uniformly sampled at 8 points and the error is computed for the 512 combinations. The maximum distance is used as the estimate for the radius of the uncertainty region. Due to the uniformity of the sample the error of this estimate is never very large. In [2], a linear approximation to the uncertainty in the image position is used to estimate the uncertainty region. The first three methods we tested are the basic method described above, the method which requires derivative estimations and the maximum of both methods.

In the fourth method we use a nonlinear minimization algorithm [8] which uses the perturbed points obtained by the third method as the starting position for the algorithm. The algorithm tries to find points on the three uncertainty circles which maximize the uncertainty in the position of the projection of  $\mathbf{m}_3$  in the image. Because in many cases the results of the third method are close to optimal, the number of pose evaluations that the algorithm needs to perform is relatively low and on average requires only 50 pose evaluations. In the fifth and sixth methods we compare the uniform sampling method presented in [3] to random sampling. In the fifth method we choose 512 random triples of points on the uncertainty circles and take the maximum displacement as the estimate. As on average this technique outperforms the method presented in [3] we reduced the number of samples until the number of times the two methods were better than each other was approximately equal. That required only 175 pose evaluations. The main drawback of the random methods is that in some rare cases the error in the estimate can be quite large where as in the uniform sampling technique the errors are relatively small.

We have tested the various methods by randomly choosing model and image point triplets and an additional model point and using the different methods to estimate the size of the uncertainty regions. Table 1 shows the cost of computing the estimate for the various methods and how often they yield a larger estimate than [3]. All the methods tested except [2] are guaranteed to produce estimates which are lower than the true value. Therefore showing this value helps compare them to [3]. Table 2 shows the percent of tests for which the relative error compared to [3] is less than a given percent.

From the results shown in Tables 1 and 2 it is clear that our methods estimate well the true uncertainty region at a fraction of the cost of [3].

However, in order to check quality of the estimates we have to compare the false positive and false negative rates of the various methods. This is done by repeating the following experiment which has been used in [2] to test their method. Sets of four model points are randomly generated and projected orthographically to a  $1000 \times 1000$  square image. The image points are then perturbed uniformly within a circle of radius five pixels. Using the first three pairs of model and image points, the uncertainty region for the fourth model point is estimated. We check to see if the fourth The translation and rotation components of the pose can be computed easily before or after the scale and viewing direction have been recovered (see for example [4, 9]). The scale has been already computed. Therefore the only component missing is the viewing direction  $\mathbf{v}$ .  $\mathbf{v}$  is a unit vector which as can be seen in Figure 1 is parallel to  $H_1$  and  $H_2$ . From that we deduce:

$$(\mathbf{m}_1 - \mathbf{m}_0) \cdot \mathbf{v} = H_1 \tag{5}$$

$$(\mathbf{m}_2 - \mathbf{m}_0) \cdot \mathbf{v} = H_2 \tag{6}$$

Each of these equations defines a plane and the intersection of these planes is a line in direction  $\mathbf{u} = (\mathbf{m}_1 - \mathbf{m}_0) \times (\mathbf{m}_2 - \mathbf{m}_0)$ . To complete the definition of the line we have to find a point p on that line. We therefore add another equation  $\mathbf{u} \cdot \mathbf{v} = \omega$  where  $\omega$ could be any number and we choose for reasons that will become apparent shortly  $\omega = 0$ . Using Gaussian elimination we solve these three linear equations yielding a point p. Thus we are looking for a point on the line  $\mathbf{v} = p + \lambda \mathbf{u}$  such that  $||p + \lambda \mathbf{u}||^2 = 1$ .

$$||p + \lambda \mathbf{u}||^2 = ||p||^2 + 2\lambda p \cdot \mathbf{u} + \lambda^2 |||\mathbf{u}||^2 = 1$$

As we chose  $\omega = 0$ , the term  $2\lambda p \cdot \mathbf{u}$  vanishes and the solutions for  $\lambda$  are:

$$\lambda = \pm \sqrt{\frac{1 - ||p||^2}{||\mathbf{u}||^2}}$$

yielding the following two viewing directions

$$\mathbf{v} = (p + \lambda \mathbf{u})$$

These viewing directions yield the given image and its reflection which can be easily separated. The viewing directions corresponding to the other pair  $(-H_1, -H_2)$  are simply  $\mathbf{v} = -(p + \lambda \mathbf{u})$ .

### 3 Uncertainty Region Estimation

We now turn to the main topic of this paper, estimating the region in the image in which the projection of a fourth model point  $\mathbf{m}_3$  will be located. We model errors by assuming that the detected feature point is within  $\epsilon$  pixels from the location of the true point. The main problem is to try to estimate the effects of the uncertainty in the image locations of  $\mathbf{i}_0$ ,  $\mathbf{i}_1$ , and  $\mathbf{i}_2$ on the estimated location of the fourth point  $\mathbf{i}_3$ . The maximum displacement occurs when the errors of the three points lie on the circle of radius  $\epsilon$ . As mentioned above the true uncertainty regions tend to be circular therefore we will try to find a combination of errors which will yield the largest displacement in the position of  $\mathbf{m}_3$  and use that as our estimate for the radius of the uncertainty region.

In [1] it is shown that the coordinates of  $\mathbf{m}_3$  can be expressed as a function of the coordinates of  $\mathbf{m}_0$ ,  $\mathbf{m}_1$ , and  $\mathbf{m}_2$  by solving the following vector equation for the "extended affine coordinates",  $(\alpha, \beta, \gamma)$ , of  $\mathbf{m}_3$ .

$$\mathbf{m}_{3} = \alpha(\mathbf{m}_{1} - \mathbf{m}_{0}) + \beta(\mathbf{m}_{2} - \mathbf{m}_{0}) + \gamma(\mathbf{m}_{1} - \mathbf{m}_{0}) \times (\mathbf{m}_{2} - \mathbf{m}_{0}) + \mathbf{m}_{0}$$
(7)

Substituting (2-4) into (7) yields that the image location of  $\mathbf{m}_3$  is

$$\begin{aligned} & (\alpha(x_1 - x_0) + \beta(x_2 - x_0) \\ & + \gamma((y_1 - y_0)H_2 - (y_2 - y_0)H_1) + x_0, \\ & \alpha(y_1 - y_0) + \beta(y_2 - y_0) \\ & + \gamma(-(x_1 - x_0)H_2 + (x_2 - x_0)H_1) + y_0) \end{aligned}$$
(8)

We will use this expression for the position of the image location of  $\mathbf{m}_3$  to estimate the uncertainty region. Let  $\epsilon_0, \epsilon_1, \epsilon_2$  be the error vectors for points  $\mathbf{i}_0, \mathbf{i}_1, \mathbf{i}_2$  respectively. Substituting the perturbed points into (8) and subtracting the unperturbed image position of  $\mathbf{m}_3$  yields:

$$(\epsilon_{0} \cdot (1 - \alpha - \beta, \gamma(H_{1} - H_{2})) + \epsilon_{1} \cdot (\alpha, \gamma H_{2}) + \epsilon_{2} \cdot (\beta, -\gamma H_{1}), \epsilon_{0} \cdot (-\gamma(H_{1} - H_{2}), 1 - \alpha - \beta) + \epsilon_{1} \cdot (-\gamma H_{2}, \alpha) + \epsilon_{2} \cdot (\gamma H_{1}, \beta))$$
(9)

 $\operatorname{Let}$ 

$$\Sigma_{0} = (1 - \alpha - \beta, \gamma(H_{1} - H_{2}))$$
  

$$\Sigma_{1} = (\alpha, \gamma H_{2})$$
  

$$\Sigma_{2} = (\beta, -\gamma H_{1})$$
  

$$\Sigma_{0}^{\perp} = (-\gamma(H_{1} - H_{2}), 1 - \alpha - \beta)$$
  

$$\Sigma_{1}^{\perp} = (-\gamma H_{2}, \alpha)$$
  

$$\Sigma_{2}^{\perp} = (\gamma H_{1}, \beta)$$

Using this notation (9) is reduced to

$$\left(\epsilon_0 \cdot \Sigma_0 + \epsilon_1 \cdot \Sigma_1 + \epsilon_2 \cdot \Sigma_2, \epsilon_0 \cdot \Sigma_0^{\perp} + \epsilon_1 \cdot \Sigma_1^{\perp} + \epsilon_2 \cdot \Sigma_2^{\perp}\right) (10)$$

Under the constraint  $||\epsilon_0|| = ||\epsilon_1|| = ||\epsilon_2|| = \epsilon$ , and neglecting for now the fact that  $H_1$  and  $H_2$  are not constants, the  $e_i$ 's which maximize the x coordinate of (10) are

$$\epsilon_i = \epsilon \frac{\Sigma_i}{||\Sigma_i||}$$

As  $\Sigma_i \cdot \Sigma_i^{\perp} = 0$  the *y* coordinate of the displacement is zero. It is easy to show that when the  $\epsilon_i$ 's computed

region are derived and the uncertainty region is estimated using linear-programming.

Our method for estimating the uncertainty regions is based on the pose estimation method presented by Alter [1] which is very efficient and has an elegant geometric interpretation. The main advantages of our method is that it is much more efficient than previous methods, yields smaller uncertainty regions which reduce the probability for false matches, and has a higher probability for the image point to be within the uncertainty region.

The rest of the paper is organized as follows. We review Alter's pose estimation method in Section 2. We present our method for estimation of the uncertainty regions using a three point match in Section 3, and extend our method to larger sets of matches in Section 4. Finally a number of issues raised by our algorithm and future research directions are discussed in Section 5.

## 2 Geometric Pose Estimation

In this section we will present a brief overview of Alter's geometric pose estimation method. For details please refer to [1]. The geometric setting underlying the weak-perspective three point pose estimation problem is shown in Figure 1. The picture shows the three model points being projected orthographically to the plane that contains  $\mathbf{m}_0$  and is parallel to the image plane, and then shows them being scaled down by scale factor s into the image. Let the distances between the model points be  $R_{01}$ ,  $R_{02}$  and  $R_{12}$ , and the corresponding distances between the image points be  $d_{01}$ ,  $d_{02}$  and  $d_{12}$ . Also let

$$\begin{split} a &= (R_{01} + R_{02} + R_{12})(-R_{01} + R_{02} + R_{12}) \\ &(R_{01} - R_{02} + R_{12})(R_{01} + R_{02} - R_{12}) \\ b &= d_{01}^2(-R_{01}^2 + R_{02}^2 + R_{12}^2) + d_{02}^2(R_{01}^2 - R_{02}^2 + R_{12}^2) \\ &+ d_{12}^2(R_{01}^2 + R_{02}^2 - R_{12}^2) \\ c &= (d_{01} + d_{02} + d_{12})(-d_{01} + d_{02} + d_{12}) \\ &(d_{01} - d_{02} + d_{12})(d_{01} + d_{02} - d_{12}) \\ \sigma &= \begin{cases} 1 & \text{if } d_{01}^2 + d_{02}^2 + d_{12}^2 \le s^2(R_{01}^2 + R_{02}^2 + R_{12}^2), \\ -1 & \text{otherwise.} \end{cases} \end{split}$$

Then the unknown parameters in Figure 1 are

$$s = \sqrt{\frac{b + \sqrt{b^2 - ac}}{a}}$$
$$(h_1, h_2) = \pm \left(\sqrt{(sR_{01})^2 - d_{01}^2}\right)$$

$$\sigma \sqrt{(sR_{02})^2 - d_{02}^2}$$
(1)  
1, H<sub>2</sub>) =  $\frac{1}{s}(h_1, h_2)$ 

For each image of three points there are two poses which yield that image. (1) yields two pairs of values for  $h_1$  and  $h_2$  which correspond to those two poses.

(H



Figure 1: Model points  $\mathbf{m}_0$ ,  $\mathbf{m}_1$ , and  $\mathbf{m}_2$  undergoing orthographic projection plus scale to produce image points  $\mathbf{i}_0$ ,  $\mathbf{i}_1$  and  $\mathbf{i}_2$  (adapted from [1]).

Given image points  $\mathbf{i}_0 = (x_0, y_0)$ ,  $\mathbf{i}_1 = (x_1, y_1)$ , and  $\mathbf{i}_2 = (x_2, y_2)$ , the 3D locations of the model points in camera-centered coordinates are:

$$\mathbf{m}_{0} = \frac{1}{s}(x_{0}, y_{0}, w) \tag{2}$$

$$\mathbf{m}_1 = \frac{1}{s} (x_1, y_1, h_1 + w) \tag{3}$$

$$\mathbf{m}_2 = \frac{1}{s}(x_2, y_2, h_2 + w) \tag{4}$$

Alter [1] did not present a method to compute the pose in model-centered coordinates from the values computed above so we will show a simple geometric method here.

# A Fast Method for Estimating the Uncertainty in the Location of Image Points in 3D Recognition<sup>\*</sup>

Ilan Shimshoni Computer Science Department Technion, Haifa 32000, Israel e-mail: ilan@cs.technion.ac.il

Abstract: Efficient and robust model-based recognition systems need to be able to estimate reliably and quickly the possible locations of other model features in the image when a match of several model points to image points is given. Errors in the sensed data lead to uncertainty in computed pose of the object which in turn lead to uncertainty in those positions. We present a very fast and accurate method for estimating these uncertainty regions. Our basic method deals with an initial match of three points and is extended using statistical methods to estimate the uncertainty region using initial matches of any size.

#### 1 Introduction

Model-based object recognition systems usually involve estimating the pose of the object using a small set of matched features. The hypothesis is then verified by using the pose to find additional matches. These systems perform the operations of pose estimation and estimation of the location of other features many times during the recognition process. Therefore efficient methods for computing them must be developed for these systems themselves to be efficient. For example, in the alignment recognition method presented by Huttenlocher and Ullman [6], the pose of the object is estimated from a match of triples of model and image points and used to predict the approximate image location of other points. If there are m model points and n image points,  $O(m^3n^3)$  pose estimations will be computed and for each pose m estimations of the position in the image of other points will be computed. Therefore it is critical for the performance of this and similar algorithms to find efficient methods for pose estimation and computation of the uncertainty region in the image of other model points.

Another important factor which is critical for the

performance of recognition algorithms is the quality of the estimate of the uncertainty region. The more exact the estimate, the lower the chances for the projection of a model point not to be within the uncertainty region (false negative). A good estimate of the position of projection of the model point will result in a smaller uncertainty region. This lowers the chances for a point in the image to be within the uncertainty region of an incorrect match (false positive), which might cause an incorrect match to be considered correct, requiring a costly verification process. Therefore a method which produces good estimates with small exact uncertainty regions will improve the efficiency of recognition algorithms considerably.

In this paper we will consider the "weakperspective" imaging model. We use this model because it is simpler than the full perspective model and because it is a good estimate of the perspective model when the object is relatively far from the camera relative to its size.

Several methods for estimating the pose of the object from a match of three model points to image points have been presented [1, 2, 4, 6, 9]. Methods for estimating the uncertainty regions have also been presented. Several methods estimate the pose uncertainty region [4, 9] which can then be used to estimate the uncertainty regions in the image, others estimate directly the uncertainty regions in the image [2, 3, 7]. Our method belongs to the latter.

In [3], the three uncertainty circles are uniformly sampled and for each combination of sampled points the position of the fourth point is calculated. The results of these experiments show that the true uncertainty regions tend to be circular. They also show that a very coarse sampling of only 8 points of each circle (i.e. 512 pose computations) is needed to achieve a very good estimate of the radius of the circle.

In [2], the uncertainty region is estimated by a linear approximation. When the given match has more than three points, linear constraints on the uncertainty

<sup>\*</sup>This work was supported in part by the Koret Foundation.