Pose Estimation from Corresponding Point Data

ROBERT M. HARALICK, FELLOW IEEE, HYONAM JOO, CHUNG-NAN LEE,
XINHUA ZHUANG, VINAY G. VAIDYA, AND MAN BAE KIM

Abstract —Solutions for four different pose estimation problems are presented. Closed form least-squares solutions are given to the over constrained 2-D–2-D and 3-D–3-D pose estimation problems. A globally convergent iterative technique is given for the 2-D perspective projection—3-D pose estimation problem. A simplified linear solution and a robust solution to the 2-D perspective projection—2-D perspective projection pose estimation problem are also given. Simulation experiments consisting of millions of trials having varying numbers of pairs of corresponding points, varying signal to noise ratio (SNR) with either Gaussian or uniform noise provide data suggesting that accurate inference of rotation and translation with noisy data may require corresponding point data sets having hundreds of corresponding point pairs when the SNR is less than 40 dB. The experiment results also show that robust technique can suppress the effect of blunder data that come from outliers or mismatched points.

I. Introduction

POSE ESTIMATION is an essential step in many machine vision problems involving the estimation of object position and orientation relative to a model reference frame or relative to the object position and orientation at a previous time using a camera sensor or a range sensor. There are four pose estimation problems with point data. Each arises from two views taken of the same object that can be thought of as having undergone an unknown rigid body motion from the first view to the second view. In model-based vision, one "view" provides three-dimensional (3-D) data relative to the model reference frame. The other is the 2-D perspective projection. In motion estimation and structure from motion problems there is a rigid body motion of the sensor, the object or both. Both views are 2-D perspective projections. In any case, in each problem corresponding point pairs from the two views are obtained from some kind of matching procedure. The pose estimation problem with corresponding point data begins with such a corresponding point data set. Its solution is a procedure that uses the corresponding point data set to estimate the translation and rotation that define the relationship between the two coordinate frames.

In the simplest pose estimation problem, the data sets consist of two-dimensional data points in a two-dimensional space. Such data sets arise naturally when flat 3-D objects are viewed under perspective projection with the look angle being the same as the surface normal of the object viewed. In the next more difficult pose estimation problem, the data sets consist of three-dimensional data points in a three-dimensional space. Such data sets arise naturally when 3-D objects are viewed with a range finder sensor. In the most difficult pose estimation problems, one data set consists of the 2-D perspective projection of 3-D points and the other data set consists of either a 3-D point data set, in which case it is known as absolute orientation problem, or the other data set consists of a second 2-D perspective projection view of the same 3-D point data set, in which case, it is known as the relative orientation problem. The latter case occurs with time-varying imagery, uncontrolled stereo or multicamera imagery.

This paper describes a solution to each of the four problems and characterizes the performance under varying conditions of noise. The simplest case is when the point positions are perturbed by independent additive Gaussian noise. Here when the signal-to-noise ratio (SNR) decreases below 40 dB, the mean error skyrockets in the more complex pose estimation problem unless there are hundreds of corresponding points pairs. Other than this phenomenon, the only interest in the additive Gaussian noise case is to establish a baseline reference against which more realistic and potentially devastating noise can be compared.

The noise having a dominant effect in point correspondence is due to incorrect matches. An incorrect match makes a point in the first view correspond to an incorrect point in the second view. Noise that models the incorrect match may be described in a variety of ways. A pair of points in one view may be incorrectly matched to a pair of points in a second view by a simple interchange. A point in one view may be matched to a point chosen at random in the second view. Or the independent additive noise may be from a distribution having tails so broad that the distribution does not have finite variance. One such distribution is the slash distribution that can be obtained as a Gaussian random variable with mean 0 and variance σ^2 divided by a uniform random variable over the unit interval $[0,1]$. the

slash density function has the form

$$f(z) = \frac{\sigma\left(1 - e^{-(1/2)(z/\sigma)^2}\right)}{z^2\sqrt{2\pi}}$$

and it is often used in characterizing the performance of robust estimators.

This paper argues that the estimators used by machine vision procedures must be robust since all machine vision feature extractors, recognizers, and matchers seem to make occasional errors which indeed are blunders. Blunders make typical estimators such as ordinary least squares estimators the estimators of least virtue. Thus it is important to pay attention to the reliability of estimators under conditions when the data has blunders.

Least-squares estimation can be made robust under blunders by converting the estimation procedure to an iterative reweighted least squares where the weight for each observation depends on its residual error and its redundancy number. It is therefore meaningful to first find the form for the least-squares solution, establish their performance as a baseline reference, put the solution technique in an iterative reweighted form, and finally evaluate the performance using nonnormal noise such as slash noise. This paper represents some initial steps in this strategy.

Section II derives a closed form least squares solution to the pure 2-D–2-D pose estimation problem. An subsequently, we derive an iterative weighted least-squares solution using a robust method. Section III derives a closed form least-squares solution to the pure 3-D–3-D pose estimation problem using a singular value decomposition technique. The least-squares solution for both the 2-D–2-D and 3-D–3-D pose estimation problems are constrained to produce rotation matrices that are guaranteed to be orthonormal. Section IV discusses an iterative solution to the 2-D perspective projection 3-D pose estimation problem. The technique appears to be globally convergent from any initial starting value. Section V discusses a solution to the 2-D perspective projection—2-D perspective projection pose estimation problem. The robust algorithm is also presented.

II. 2-D–2-D ESTIMATION

There are a variety of model-based inspection tasks that require the coordinate system of an object model to be aligned with the coordinate system of a set of observations before the actual inspection judgements can be made. One example is surface mount device inspection on printed circuit boards. Here the image processing produces, among other measurements, the observed center position of each device. The model stores, in the printed circuit board coordinate system, the center positions, orientations, and sizes of all devices. To determine whether each device that should be present is present, and whether everything observed to be present is actually present and in its correct position and orientation first requires determining the relationship between the coordinate system of the observed image and the coordinate system of the model. Usually

this relationship is given by a 2-D rotation and translation.

As mentioned in Section I, in the matching process, the noise is a big factor that disturbs the pose estimation. The noise of a great concern is incorrect matching of the data points. The incorrect match makes a data point of the model to correspond to an incorrect point of the image. (These incorrect points will be called "outliers" through the paper.) The outliers may affect the accuracy and stability of the pose estimation.

We have recognized that some data points, which arise from heavily tailed distributions or are simply bad sample data points due to errors, degrade the performance and accuracy of the least-squares approach. The estimated parameter values may be useless or unreliable in the presence of such erroneous data points. Therefore we need a new method to weaken the effect of the outliers and then to improve the performance and reliability of the least-squares method.

For the purpose of removing the outliers from the pose estimation, we make use of a robust method. The robust method has been developed to modify the least-squares method so that the outliers have much less influence on the final estimates. Since the outliers are eliminated or weakened, the estimation of the 2-D pose will be more accurate, reliable and stable.

The section of 2-D–2-D pose estimation is organized as follows. Section II-A gives a precise statement of this problem as a weighted least-squares problem. In Section II-B, we introduce a derivation of the solution using the least-squares method. In subsequent sections we introduce the robust method using an iterative weighted least-squares method. In Section II-D, we present numerical results of the two methods and discuss the performances of them. From the numerical results we conclude that the robust method produces a better and more stable performance than the least-squares method in the 2-D–2-D pose estimation.

A. Statement of Problem

In the simple two-dimensional pose detection problem, we are given N two-dimensional coordinate observations from the observed image: $x_1, \cdots, x_N$. These could correspond, for example, to the observed center position of all observed objects. We are also given the corresponding or matching N two-dimensional coordinate vectors from the model: $y_1, \cdots, y_N$. In the usual inspection situation, establishing which observed vector corresponds to which model vector is simple because the object being observed is fixtured and its approximate position and orientation are known. The approximate rotational and translational relationship between the image coordinate system and the object coordinate system permits the matching to be done just by matching a rotated and translated image position to an object position. The match is established if the rotated image position is close enough to the object position.

In the ideal case, the simple 2-D pose detection problem is to determine from the matched points a more precise estimate of a rotation matrix R and a translation t such that $y_n = Rx_n + t$, $n = 1, \cdots, N$. Since there are likely to be small observational errors, the real problem must be posed as a minimization. Determine R and t that minimize the weighted sum of the residual errors ϵ^2

$$\epsilon^2 = \sum_{n=1}^{N} w_n \| y_n - (Rx_n + t) \|^2. \tag{1}$$

The weights w_n, $n = 1, \cdots, N$ satisfy $w_n \geqslant 0$ and $\sum_{n=1}^{N} w_n = 1$. If there is no prior knowledge as to how the weights should be set, they can be defined to be equal: $w_n = 1/N$. Otherwise they can be set to $1/6_n^2$ if the variances of the observations are known.

B. Least-Squares Method

Upon expanding (1) out we have

$$\epsilon^2 = \sum_{n=1}^{N} w_n [(y_n - t)'(y_n - t) - (y_n - t)' Rx_n$$
$$- x_n' R'(y_n - t) + x_n' R' Rx_n]. \tag{2}$$

Since R is a rotation matrix, it is orthonormal so that $R^{-1} = R'$. Also since $(y_n - t)' Rx_n$ is a scalar it is equal to its transpose. Hence

$$\epsilon^2 = \sum_{n=1}^{N} w_n [(y_n - t)'(y_n - t) - 2(y_n - t)' Rx_n + x_n' x_n]. \tag{3}$$

Taking the partial derivative of ϵ^2 with respect to the components of the translation t and setting the partial derivative to 0, we obtain

$$0 = \sum_{n=1}^{N} w_n [-2(y_n - t) + 2Rx_n]. \tag{4}$$

Letting

$$\bar{x} = \sum_{n=1}^{N} w_n x_n \qquad \bar{y} = \sum_{n=1}^{N} w_n y_n \tag{5}$$

there immediately results

$$\bar{y} = R\bar{x} + t. \tag{6}$$

Substituting $\bar{y} - R\bar{x}$ for t in the expression for the residual error we can do some simplifying

$$\epsilon^2 = \sum_{n=1}^{N} w_n [(y_n - (\bar{y} - R\bar{x}))'(y_n - (\bar{y} - R\bar{x}))$$
$$- 2(y_n - (\bar{y} - R\bar{x}))' Rx_n + x_n' x_n]$$
$$= \sum_{n=1}^{N} w_n [(y_n - \bar{y})'(y_n - \bar{y}) + 2(y_n - \bar{y})' R\bar{x} + \bar{x}' R' R\bar{x}$$
$$- 2(y_n - \bar{y})' Rx_n - 2\bar{x} R' Rx_n + x_n' x_n]$$
$$= \sum_{n=1}^{N} w_n [(y_n - \bar{y})'(y_n - \bar{y}) - 2(y_n - \bar{y})' R(x_n - \bar{x})$$
$$+ (x_n - \bar{x})'(x_n - \bar{x})]. \tag{7}$$

The counterclockwise rotation angle θ is related to the rotation matrix by

$$R = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \tag{8}$$

We want to take the partial derivative of ϵ^2 with respect to θ. Now we need a notation in which the two components of x_n and the two components of y_n can be written explicitly. Letting

$$x_n = \begin{pmatrix} x_{1n} \\ x_{2n} \end{pmatrix}, \qquad y_n = \begin{pmatrix} y_{1n} \\ y_{2n} \end{pmatrix}$$
$$\bar{x} = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix}, \qquad \bar{y} = \begin{pmatrix} \bar{y}_1 \\ \bar{y}_2 \end{pmatrix}. \tag{9a}$$

Then

$$(y_n - \bar{y})' R(x_n - \bar{x}) = (y_{n1} - \bar{y}_1) \cos\theta (x_{n1} - \bar{x}_1)$$
$$+ (y_{n1} - \bar{y}_1)(-\sin\theta)(x_{n2} - \bar{x}_2)$$
$$+ (y_{n2} - \bar{y}_2) \sin\theta (x_{n1} - \bar{x}_1)$$
$$+ (y_{n2} - \bar{y}_2) \cos\theta (x_{n2} - \bar{x}_2). \tag{9b}$$

Setting to zero the partial derivative of ϵ^2 with respect to θ results in

$$0 = -2 \sum_{n=1}^{N} w_n [(y_{n1} - \bar{y}_1)(-\sin\theta)(x_{n1} - \bar{x}_1)$$
$$+ (y_{n1} - \bar{y}_1)(-\cos\theta)(x_{n2} - \bar{x}_2)$$
$$+ (y_{n2} - \bar{y}_2) \cos\theta (x_{n1} - \bar{x}_1)$$
$$+ (y_{n2} - \bar{y}_2)(-\sin\theta)(x_{n2} - \bar{x}_2)], \tag{10}$$

letting

$$A = \sum_{n=1}^{N} w_n [(y_{n1} - \bar{y}_1)(x_{n1} - \bar{x}_1) + (y_{n2} - \bar{y}_2)(x_{n2} - \bar{x}_2)]$$

$$B = \sum_{n=1}^{N} w_n [(y_{n1} - \bar{y}_1)(x_{n2} - \bar{x}_2) - (y_{n2} - \bar{y}_2)(x_{n1} - \bar{x}_1)].$$

Then

$$0 = A \sin\theta + B \cos\theta. \tag{11}$$

Hence

$$\cos\theta = \frac{-A}{\sqrt{A^2 + B^2}} \qquad \sin\theta = \frac{B}{\sqrt{A^2 + B^2}} \tag{12}$$

or

$$\cos\theta = \frac{A}{\sqrt{A^2 + B^2}} \qquad \sin\theta = \frac{-B}{\sqrt{A^2 + B^2}}. \tag{13}$$

The correct value for θ will in general be unique and will be that θ that minimizes ϵ^2. Thus the better of the two

choices can always be easily determined by simply substituting each value for θ into the original expression for ϵ^2.

In this subsection, we assumed that w_n is given. To remove or lessen the effect of the outliers and thereby improve the performance and stability of the pose estimation, the weights need to be determined based on the data. For this we need a method to assign a weight based on the residual error. The outliers are forced to have small or zero weights, lessening their effect on the pose estimation. It is also reasonable that the data points with small noise are assigned larger weights than those with large noise error. From this assumption, we may expect better performance and stability in the pose estimation. The method to assign appropriate weights to the data points is done by a robust method using an iterative weighted least-squares method, which is described in the next subsection.

C. Robust Method

In the previous subsection, we have presented the weighted least-squares method where the weights are given. In this subsection we will introduce an iterative weighted least-squares method where the weights are data dependent. The purpose is to make the outliers have zero or small weights and thus to eliminate the effects of them in the pose estimation.

1) M-Estimator: In *M*-estimator, the solution for θ is given by a minimization problem of the following form

$$\min_{\theta} \sum_{i=1}^{N} \rho(x_i - \theta) \tag{14}$$

or by an implicit equation

$$\sum_{i=0}^{N} \psi(x_i - \theta) = 0 \tag{15}$$

where N is the sample size. The ρ is an arbitrary nonnegative monotonically increasing function (called the object function) for positive argument and monotonically decreasing for negative argument. The $\psi(x_i - \theta)$ is a derivative of $\rho(x_i - \theta)$ with respect to θ and is called an *M*-estimator shown as

$$\psi(x_i - \theta) = \frac{\partial}{\partial \theta} \rho(x_i - \theta). \tag{16}$$

Equation (15) can be written equivalently as

$$\sum_{i=0}^{N} w_i(x_i - \theta) = 0 \tag{17}$$

where

$$w_i = \frac{\psi(x_i - \theta)}{x_i - \theta}, \qquad i = 1, \cdots, N. \tag{18}$$

This gives a formal representation of θ as a weighted mean

$$\theta = \frac{\sum\limits_{i=1}^{N} w_i x_i}{\sum\limits_{i=1}^{n} w_i} \tag{19}$$

with weights depending on the data.

Among many forms of functions ρ and ψ proposed in the literature, Tukey's form is investigated in this experiment. The Tukey's biweight $\psi(x)$ is

$$= \begin{cases} x\left[1 - \left(\dfrac{x}{cS}\right)^2\right]^2, & \text{if } |x| \leqslant cS; \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

The c is a tuning constant that typically lies in the range 6–12. In the experiments we adopted 6 as a value of c. S is a scale estimator that is usually MAD (median of absolute deviation). The cS is called "rejection point."

The corresponding object function of the Tukey's biweight, $\rho(x)$ is

$$\rho(x) = \begin{cases} 1/6\left[1 - \left(1 - \left(\dfrac{x}{cS}\right)^2\right)^3\right], & \text{if } |x| \leqslant cS; \\ 1/6, & \text{otherwise.} \end{cases} \tag{21}$$

The weight function of the Tukey's biweight is

$$w(x) = \begin{cases} \left[1 - \left(\dfrac{x}{cS}\right)^2\right]^2, & \text{if } |x| \leqslant cS; \\ 0, & \text{otherwise.} \end{cases} \tag{22}$$

Since it is difficult to find a closed form for the estimated parameter θ, an iterative method is usually used.

2) Iterative Weighted Least-Squares Method: The residual error ϵ_i for nth data sample is

$$\epsilon_i = y_i - (Rx_i + t)$$

where $i = 1, \cdots, N$. N is a sample size. The robust estimation procedure is implemented as the following iterative method. Given the data sets x_i and y_i, where $i = 1, \cdots, N$.

- Select initial starting values for R and t.
- R and t give weights w_i where $i = 1, \cdots, N$. To find weights, we use (22). x^2 is replaced by the residual error, $\|\epsilon_i\|^2$, where $\epsilon_i = y_i - x_i R' - E_i t$ and $i = 1, \cdots, N$. Thus, w_i is expressed as

$$w_i = \begin{cases} \left[1 - \dfrac{\|\epsilon_i\|^2}{(cS)^2}\right]^2, & \text{if } \|\epsilon_i\| \leqslant cS; \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

The new R and t are obtained from the new weights.
- If some degree of convergence in R and t are obtained, go to the next step. If not go back one step.
- From the final W, we normalize the weights and find estimates for rotation angle and translation using the solution derived in Section II-B. Stop.
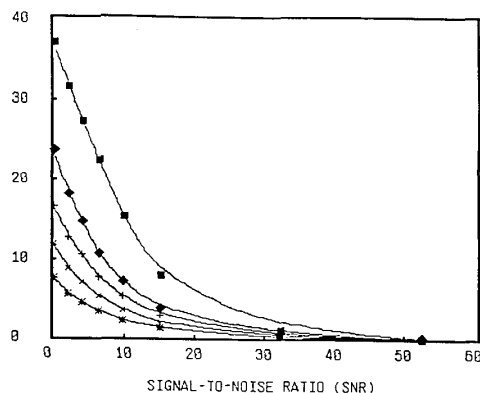
Fig. 1. Mean absolute rotational error as function of SNR for 2-D–2-D pose estimation problem. ■, $N = 8$; ◆, $N = 25$; +, $N = 50$; ×, $N = 100$; ∗, $N = 200$.
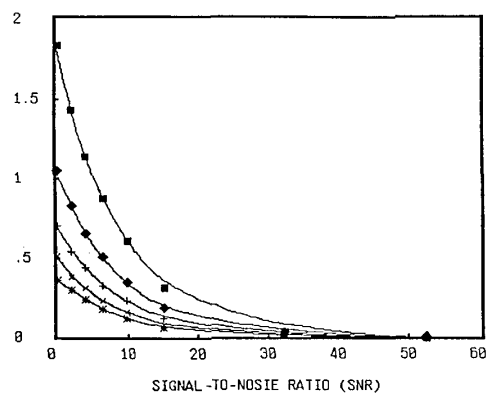

Fig. 2. Mean translational distance error as function of SNR for 2-D–2-D pose estimation problem. ■, $N = 8$; ◆, $N = 25$; +, $N = 50$; ×, $N = 100$; ∗, $N = 200$.

D. Experimental Results

For each trial, object data points were generated uniformly in the square $[-2,2] \times [-2,2]$. A rotation angle was chosen from the interval $[-15,15]$ (in degrees) according to a uniform distribution and the translation vector was chosen from the square $[-1,1] \times [-1,1]$ also according to a uniform distribution. Independent Gaussian noise was added to the rotated and translated points and the SNR, defined as 20 log peak-to-peak signal/normalized interquartile range, was varied between 0 dB and 52 dB. The normalized interquartile range is defined as the interquartile range of the noise divided by the interquartile range of a Gaussian variate having variance 1. For noise that is Gaussian the normalized interquartile range is just the noise standard deviation. For distributions such as the slash distribution that does not have finite variance, the normalized interquartile range is a suitable estimate of dispersion. For each different combination of SNR and number of corresponding point pairs, one thousand trials were made. First we made experiments without generating any outliers and examined the performance of the least-squares method. The results are shown in Figs. 1 and 2. Fig. 1 shows the mean absolute error of the rotation angle as a function of SNR for number of corresponding point pairs varying between 8 and 200. For number of corresponding point pairs equal to 8, the SNR must exceed 40 dB to guarantee mean absolute error of less than one degree while for 100 corresponding point pairs the SNR can go as low as 25 dB while maintaining a less than one degree mean absolute rotation error. The pattern for mean translational distance error is similar. This is shown in Fig. 2. To maintain a mean translational distance error of 0.01, which is a relative error of about 0.25 percent, requires 100 corresponding point pairs at a 32-dB SNR. Using only 8 corresponding point pairs, even an SNR of 52 dB provides a mean translation distance error of about 0.03, or 0.75 percent.

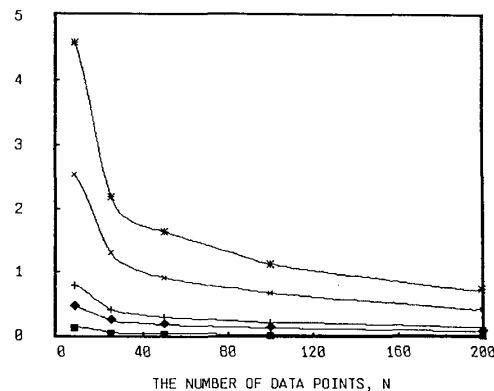In the next experiments, we examine the performance of the least-squares and robust methods with outliers present

Fig. 3. Mean absolute rotational error of least-squares method and robust method as a function of SNR for 2-D–2-D pose estimation problem. Number of corresponding point pairs is 20. Percentage of outliers is changed. ■, SNR = 50 dB; ◆, SNR = 40 dB; +, SNR = 35 dB; ×, SNR = 25 dB; ∗, SNR = 20 dB.

in the image. To generate the outliers, we intentionally changed the positions of some data points by randomly selecting arbitrary positions in the image generated according to a uniform distribution. We applied the least-squares and robust methods to estimate the pose and observed the performance. The percentage of the ouliers was varied from 10 percent to 50 percent. Figs. 3 and 4 show the mean rotational and translational errors as a function of the SNR for the PO (percentage of the outliers) varying between 10 percent and 50 percent when the least-squares and robust methods are used. The number of corresponding point pairs is 20. As we increase the PO, the performance is degraded. The robust method shows better performance than the least-squares method when the SNR is greater than 10 dB. If the SNR is less than 10 dB, the performances of the two methods are almost identical. This indicates that below 10 dB, there is not enough consistency within the data to enable a distinction between outliers and nonoutliers.

III. 3-D-3-D Estimation

A. Statement of Problem

Let $y_1, \cdots, y_N$ be N points in Euclidean 3-space. Let R be a rotation matrix and t be a translation vector. Let $x_1, \cdots, x_N$ be the points in Euclidean 3-space that match $y_1, \cdots, y_N$. Each x_n is the same rigid body motion of y_n. Hence each y_n is obtained as a rotation of x_n plus a translation plus noise.

$$y_n = Rx_n + t + \eta_n. \tag{24}$$

The 3-D-3-D pose estimation problem is to infer R and t from $x_1, \cdots, x_N$ and $y_1, \cdots, y_N$.

B. Derivation

To determine R and t we set up a constrained least-squares problem. We will minimize

$$\sum_{n=1}^{N} w_n \| y_n - (Rx_n + t) \|^2$$

subject to the constraint that R is a rotation matrix, that is, $R' = R^{-1}$. To be able to express these constraints using Lagrangian multipliers we let

$$R = \begin{pmatrix} r_1' \\ r_2' \\ r_3' \end{pmatrix}$$

where each r_i is a 3×1 vector.

The constraint $R' = R^{-1}$, then amounts to the six constraint equations

$$r_1' r_1 = 1$$
$$r_2' r_2 = 1$$
$$r_3' r_3 = 1$$
$$r_1' r_2 = 0$$
$$r_1' r_3 = 0$$
$$r_2' r_3 = 0. \tag{25}$$

The least-squares problem with constraints given by (25) can be written as minimizing ϵ^2 where

$$\epsilon^2 = \sum_{n=1}^{N} \sum_{k=1}^{3} w_n (y_{nk} - r_k' x_n - t_k)^2 + \sum_{k=1}^{3} \lambda_k (r_k' r_k - 1)$$

$$+ 2\lambda_4 r_1' r_2 + 2\lambda_5 r_1' r_3 + 2\lambda_6 r_2' r_3$$

$$x_n = \begin{pmatrix} x_{n1} \\ x_{n2} \\ x_{n3} \end{pmatrix} \qquad y_n = \begin{pmatrix} y_{n1} \\ y_{n2} \\ y_{n3} \end{pmatrix} \qquad t = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix}. \tag{26}$$

Taking the partial derivative of ϵ^2 with respect to t_n, there results

$$\frac{\partial \epsilon^2}{\partial t_k} = \sum_{n=1}^{N} 2w_n (y_{nk} - r_k' x_n - t_k)(-1), \qquad k = 1, 2, 3.$$
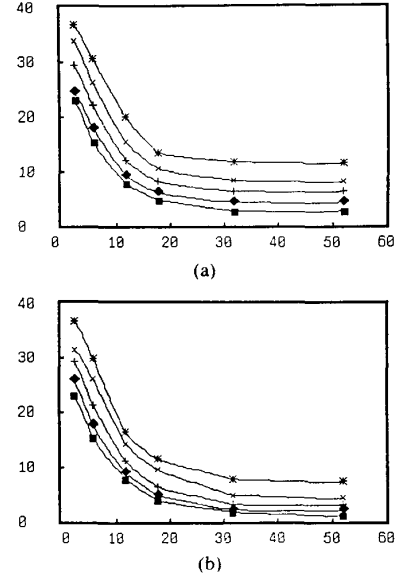

Fig. 4. Mean translational distance error of (a) least-squares method and (b) robust method as function of SNR for 2-D-2-D pose estimation problem. Number of corresponding point pairs is 20. Percentage of outliers is: ■ is 10 percent. ◆ is 20 percent. + is 30 percent. × is 40 percent. * is 50 percent.

Setting these partials to zero results in

$$\sum_{n=1}^{N} w_n (y_n - Rx_n - t) = 0.$$

By rearranging we obtain

$$t = \bar{y} - R\bar{x} \tag{27}$$

where

$$\bar{x} = \frac{\sum_{n=1}^{N} w_n x_n}{\sum_{n=1}^{N} w_n} \qquad \bar{y} = \frac{\sum_{n=1}^{N} w_n y_n}{\sum_{n=1}^{N} w_n}.$$

Thus once R is known, t is quickly determined from (27). Substituting $\bar{x} - R\bar{y}$ for t in the definition of ϵ^2, there results

$$\epsilon^2 = \sum_{n=1}^{N} w_n \sum_{k=1}^{3} (y_{nk} - \bar{y}_n - r_k'(x_n - \bar{x}))^2 + \sum_{k=1}^{3} \lambda_k (r_k' r_k - 1)$$

$$+ 2\lambda_4 r_1' r_2 + 2\lambda_5 r_1' r_3 + 2\lambda_6 r_2' r_3 \tag{28}$$

where

$$\bar{x} = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \end{pmatrix} \qquad \bar{y} = \begin{pmatrix} \bar{y}_1 \\ \bar{y}_2 \\ \bar{y}_3 \end{pmatrix}.$$

Now we take partial derivatives of ϵ^2 with respect to the components of each y_n. To write things more compactly, by $\partial \epsilon^2 / \partial r_n$ we mean a 3×1 vector whose components are the partial derivative of ϵ^2 with respect to each of the components of r_n. Then

$$\frac{\partial \epsilon^2}{\partial r_1} = \sum_{n=1}^{N} 2 w_n \left(y_{n1} - \bar{y}_1 - r_1'(x_n - \bar{x}) \right)(x_n - \bar{x})(-1)$$

$$+ 2\lambda_1 r_1 + 2\lambda_4 r_2 + 2\lambda_5 r_3 \qquad (29)$$

$$\frac{\partial \epsilon^2}{\partial r_2} = \sum_{n=1}^{N} 2 w_n \left(y_{n2} - \bar{y}_2 - r_2'(x_n - \bar{x}) \right)(x_n - \bar{x})(-1)$$

$$+ 2\lambda_2 r_2 + 2\lambda_4 r_1 + 2\lambda_6 r_3 \qquad (30)$$

$$\frac{\partial \epsilon^2}{\partial r_3} = \sum_{n=1}^{N} 2 w_n \left(y_{n3} - \bar{y}_3 - r_3'(x_n - \bar{x}) \right)(x_n - \bar{x})(-1)$$

$$+ 2\lambda_3 r_3 + 2\lambda_5 r_1 + 2\lambda_6 r_2. \qquad (31)$$

Setting these partial derivatives to zero and rearranging we obtain

$$\sum_{n=1}^{N} w_n (x_n - \bar{x})(x_n - \bar{x})' r_1 + \lambda_1 r_1 + \lambda_4 r_2 + \lambda_5 r_3$$

$$= \sum_{n=1}^{N} w_n (y_{n1} - \bar{y}_1)(x_n - \bar{x}) \qquad (32)$$

$$\sum_{n=1}^{N} w_n (x_n - \bar{x})(x_n - \bar{x})' r_2 + \lambda_4 r_1 + \lambda_2 r_2 + \lambda_6 r_3$$

$$= \sum_{n=1}^{N} w_n (y_{n2} - \bar{y}_2)(x_n - \bar{x}) \qquad (33)$$

$$\sum_{n=1}^{N} w_n (x_n - \bar{x})(x_n - \bar{x})' r_3 + \lambda_5 r_1 + \lambda_6 r_2 + \lambda_3 r_3$$

$$= \sum_{n=1}^{N} w_n (y_{n3} - \bar{y}_3)(x_n - \bar{x}). \qquad (34)$$

Let

$$A = \sum_{n=1}^{N} (x_n - \bar{x})(x_n - \bar{x})'$$

$$\Lambda = \begin{pmatrix} \lambda_1 & \lambda_4 & \lambda_5 \\ \lambda_4 & \lambda_2 & \lambda_6 \\ \lambda_5 & \lambda_6 & \lambda_3 \end{pmatrix}$$

and

$$B = (b_1 b_2 b_3)$$

where

$$b_k = \sum_{n=1}^{N} w_n (y_{nk} - \bar{y}_k)(x_n - \bar{x}).$$

Then (32), (33), and (34) can be simply rewritten as

$$AR' + R'\Lambda = B. \qquad (35)$$

Multiplying both sides of (35) on the left by R we have

$$RAR' + \Lambda = RB. \qquad (36)$$

Since $A = A'$, $(RAR')' = RAR'$. Since both RAR' and Λ are symmetric, the left-hand side must be symmetric. Hence the right-hand side is also symmetric. This means
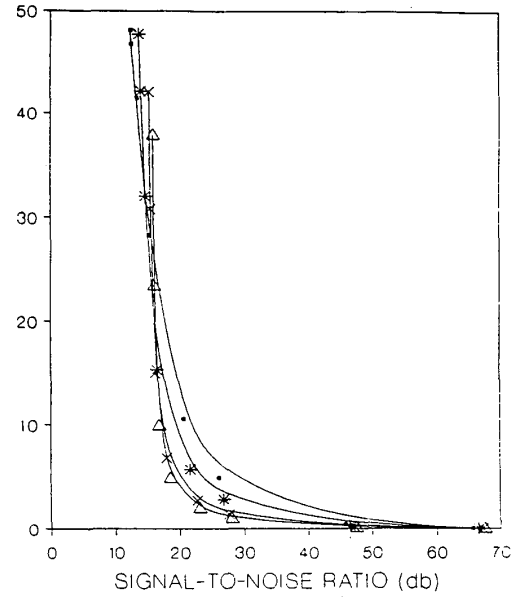
$$RB = (RB)'. \qquad (37)$$

Fig. 5. Mean rotation angle error versus SNR with Gaussian noise. Corresponding point data set sizes are — • 10, — * 25, — X 100, — △ 200. Each point on graph represents 1000 trials.

The solution for R now comes quickly. Let the singular value decomposition of B be

$$B = UDV$$

where U and V are orthonormal and D is diagonal. Then

$$RUDV = (UDV)' R'$$

$$= V'DU'R'. \qquad (38)$$

By observation, a solution for R is immediately obtained as

$$R = V'U'. \qquad (39)$$

Solutions to this problem can be found in the photogrammetry literature beginning with Thompson [18], Schut [14], Tienstra [20], and Pope [11]. Blais [1] gives a solution to the problem in the case where there may be a scale factor or magnification different than 1. Sansò [13] gives a solution to the problem using quaternions. Arun et al. [2] and Haralick et al. [25] have discussed the singular value decomposition approach to the problem.

C. Experimental Results

Over 144,000 simulation experiments were done in which 3-D points were chosen at random. A random rotation and translation are chosen and a corresponding point data set was created by rotating and translating the initial set of points and adding noise as given in (24). The rotation and translation was then estimated using (27) and (39).

The number of corresponding point pairs was varied between 10 and 200 in nine steps. The signal-to-noise ratio, which is defined as 20 log (dynamic range of 3-D points/normalized interquartile range of noise), was varied between Gaussian and Uniform. For each calculation one thousand trials were run.

Fig. 5 illustrates a typical experimental result. It shows the mean angle error of the rotation, in degrees, as a function of SNR with Gaussian noise. The plot indicates that when the number of 3-D points is 50, then the RMS error of the rotation angle will be less than 3 degrees when the SNR is greater than 55 dB.

Fig. 6 shows rotation angle error plotted as a function of number of points in the corresponding point data sets for varying levels of Gaussian noise. This plot clearly shows that when the number of corresponding point data pairs is below 40, the estimated values are unreliable. When the number of corresponding point data pairs is above 40, the estimates improve for increasing-sized sets.

The plot of the translation error angle as a function of the number of corresponding point data pairs for varying SNR and Gaussian noise is similar.

IV. 2-D PERSPECTIVE PROJECTION–3-D POSE ESTIMATION

Let $y_1, \cdots, y_N$ be the observed 3-D model points in Euclidean 3-space. Let R be a rotation matrix and t be a translation vector. Let (u_{n1}, u_{n2}), $n = 1, \cdots, N$ be the corresponding 2-D perspective projection of the 3-D points. Then the relationship between the 3-D model points and the 2-D perspective projection points is given by

$$u_{n1} = f \frac{r_1 y_n + t_1}{r_3 y_n + t_3}$$

$$u_{n2} = f \frac{r_2 y_n + t_2}{r_3 y_n + t_3}$$

$$t = (t_1, t_2, t_3)'$$

$$R = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} \tag{40}$$

where f, the focal length, is the distance of the image plane in front of the origin that is the center of perspectivity. In the 3-D coordinate system of the camera, the perspective projections are given by

$$u_n = \begin{pmatrix} u_{n1} \\ u_{n2} \\ f \end{pmatrix} = f \begin{pmatrix} v_{n1} \\ v_{n2} \\ 1 \end{pmatrix} = f v_n \tag{41}$$

where $u_{n1} = f v_{n1}$ and $u_{n2} = f v_{n2}$.

The problem of pose estimation is to determine the unknown rotation matrix R and the translation vector t given the 3-D model points and the corresponding 2-D perspective projection points on the image plane. This problem is known as the exterior orientation problem in the photogrammetry literature. The dissertation by Szczepanski [17] surveys nearly 80 different solutions beginning with one given by Schrieber of Karlsruhe in the year 1879. The first robust solution in the computer vision literature was Fischler and Bolles [4]. Wrobel and Klemm [22] discuss the fact that there are configurations of points for which the solution is unstable.

A. Iterative Least-Squares Solution

This section describes iterative procedures for determining a least-squares solution for R and t. In the following subsections we use the superscript or subscript k to denote the values in the kth iteration step. Let

$$x_n = \begin{pmatrix} x_{n1} \\ x_{n2} \\ x_{n3} \end{pmatrix} = R \begin{pmatrix} y_{n1} \\ y_{n2} \\ y_{n3} \end{pmatrix} + t \tag{42}$$

be the rotated and translated point of y_n. Let d_n be the estimated depth of each point x_n relative to the camera coordinate system.

1) Method 1: One iterative procedure for determining a least-squares solution for R and t is the following.

1) Choose initial reasonable values for the depth d_n^0 of each point. The initial values could, for example, be the same constant for each point, the constant representing an initial guess of how far the object is from the perspective center.

2) Iterate. Suppose the depth values d_n^k, $n = 1, \cdots, N$ are given. Define the depth values for the $(k+1)$th iteration by:

  a) Find the rotation matrix R_k and the translation vector t_k that minimizes

$$\epsilon_k^2 = \sum_{n=1}^{N} w_n \left\| R_k y_n + t_k - d_n^k v_n \right\|^2 \tag{43}$$

  where the $\{ w_n \mid n = 1, \cdots, N \}$ are nonnegative weights reflecting the goodness of the observations. R_k and t_k constitute the solution to the 3-D–3-D pose estimation problem.

  b) Define

$$d_n^{k+1} = \left(\frac{D_y}{D_x} \right) x_{n3}^k$$

  where

$$\bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n \qquad \bar{y} = \frac{1}{N} \sum_{n=1}^{N} y_n$$

  and

$$D_y = \sum_{n=1}^{N} \left\| y_n - \bar{y} \right\|^2$$

$$D_x = \sum_{n=1}^{N} \left\| x_n - \bar{x} \right\|^2. \tag{44}$$

A typical convergence characteristic of the computed depth values is shown in Fig. 7. This experiment is performed in a noise-free environment with $N = 10$. The depth values of the first five points are plotted against the iteration number. The correct depth values are 33.27, 34.98, 38.81, 40.39, and 42.68.

2) Method 2: Replace the Step 2b) of Method 1 with Step 1) of Method 2.

1) Define d_n^{k+1} by

$$d_n^{k+1} = \frac{(R_k y_n + t_k)' v_n}{v_n' v_n}. \quad (45)$$

It can be shown that $\epsilon_{k+1}^2 \leqslant \epsilon_k^2$ and

$$\epsilon_{k+1}^2 = \sum_{n=1}^{N} w_n \| R_{k+1} y_n + t_{k+1} - d_n^{k+1} v_n \|^2$$

$$\leqslant \sum_{n=1}^{N} w_n \| R_k y_n + t_k - d_n^{k+1} v_n \|^2$$

$$= \sum_{n=1}^{N} w_n \| (x_n^k - d_n^k v_n) + (d_n^k v_n - d_n^{k+1} v_n) \|^2$$

$$= \sum_{n=1}^{N} w_n \Big[\| (x_n^k - d_n^k v_n) \|^2 + 2(x_n^k - d_n^k v_n)'$$

$$\cdot (d_n^k - d_n^{k+1}) v_n + (d_n^k - d_n^{k+1})^2 \| v_n \|^2 \Big]$$

$$= \epsilon_k^2 + \sum_{n=1}^{N} w_n (d_n^k - d_n^{k+1})$$

$$\cdot \Big[2(x_n^k - d_n^k v_n)' v_n + (d_n^k - d_n^{k+1}) \| v_n \|^2 \Big]$$

$$= \epsilon_k^2 + \sum_{n=1}^{N} w_n (d_n^k - d_n^{k+1})$$

$$\cdot \Big[2 x_n^{k\prime} v_n - 2 d_n^k \| v_n \|^2 + (d_n^k - d_n^{k+1}) \| v_n \|^2 \Big]$$

$$= \epsilon_k^2 + \sum_{n=1}^{N} w_n (d_n^k - d_n^{k+1})$$

$$\cdot \Big[2 x_n^{k\prime} v_n - (d_n^k - d_n^{k+1}) \| v_n \|^2 \Big]$$

$$= \epsilon_k^2 + \sum_{n=1}^{N} w_n \| v_n \|^2 \Big[(d_n^{k+1})^2 - 2 \frac{x_n^{k\prime} v_n}{\| v_n \|^2} d_n^{k+1}$$

$$+ 2 \frac{x_n^{k\prime} v_n}{\| v_n \|^2} d_n^k - (d_n^k)^2 \Big]$$

$$= \epsilon_k^2 + \sum_{n=1}^{N} w_n \| v_n \|^2 \Big[\left(d_n^{k+1} - \frac{x_n^{k\prime} v_n}{\| v_n \|^2} \right)^2$$

$$- \left(d_n^k - \frac{x_n^{k\prime} v_n}{\| v_n \|^2} \right)^2 \Big]. \quad (46)$$

Consider the terms in the bracket as a function of d_n^{k+1}. The function reaches a minimum when

$$d_n^{k+1} = \frac{x_n^{k\prime} v_n}{\| v_n \|^2}. \quad (47)$$

The resulting value of the terms in the bracket at the minimum is

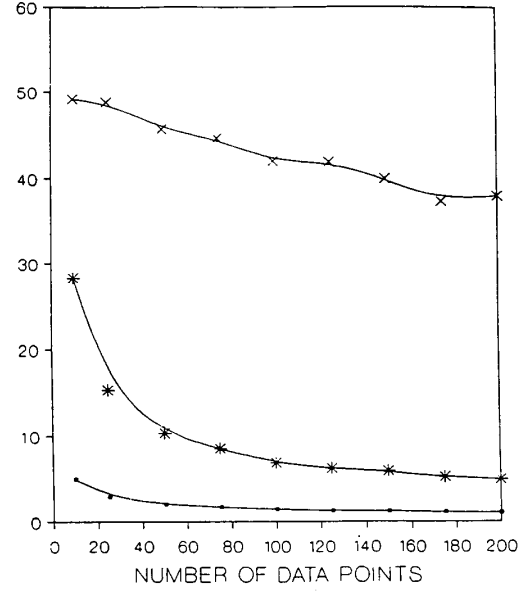$$- \left(d_n^k - \frac{x_n^{k\prime} v_n}{\| v_n \|^2} \right)^2. \quad (48)$$

Fig. 6. Mean rotation angle error versus number of points with Gaussian noise.   — · SNR 27 dB,   — * SNR 18 dB,   — X SNR 15 dB.
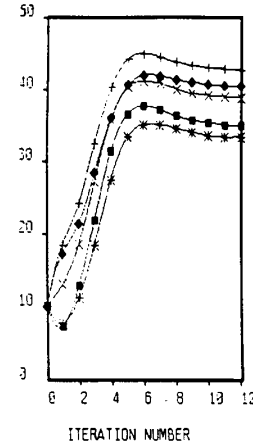

ITERATION NUMBER

Fig. 7. Illustrates convergence characteristics of Method 1. Convergence is achieved in about ten iterations. ■ point 1, ◆ point 2, + point 3, X point 4, * point 5.

This value cannot be positive. Since $w_n \| v_n \|^2 > 0$, when

$$d_n^{k+1} = \frac{x_n^{k\prime} v_n}{\| v_n \|^2} \quad (49)$$

each term in the summation is not positive and from this we can infer

$$\epsilon_{k+1}^2 \leqslant \epsilon_k^2. \quad (50)$$

A typical convergence characteristic of the computed depth values is shown in Fig. 8. This experiment is performed in a noise-free environment with $N = 10$. The depth values of the first five points are plotted against the iteration number. Notice how the convergence is monotonic. The correct depth values are 33.27, 34.98, 38.81, 40.39, and 42.68.
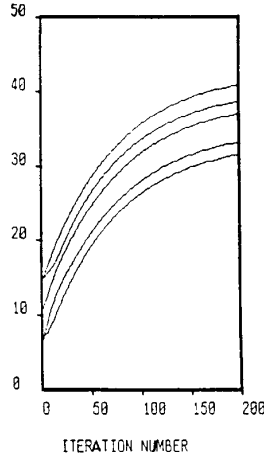
Fig. 8. Illustrates convergence characteristics of Method 2. Convergence has been observed to be monotonic and is achieved in few hundred iterations.

B. Least-Squares Adjustment by Linearization

Let ϕ, θ, and ψ be the three angles that define the rotation matrix R such that

$$R = R_x(\phi)R_y(\theta)R_z(\psi)$$

$$= \begin{pmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & \sin\phi\cos\theta \\ \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & \cos\phi\cos\theta \end{pmatrix}.$$

As there always exists random errors in the measurement of the image coordinates, let

$$u_{ni} = u_{ni}^0 + v_{ni}, \qquad i = 1,2, \qquad n = 1,\cdots,N \qquad (51)$$

where (u_{n1}^0, u_{n2}^0) are the measured image points and (v_{n1}, v_{n2}) are the corrections needed to account for the random error in the measured coordinates. Similarly, let

$$\phi = \phi^0 + \Delta\phi$$
$$\theta = \theta^0 + \Delta\theta$$
$$\psi = \psi^0 + \Delta\psi$$
$$t_i = t_i^0 + \Delta t_i, \qquad i = 1,2,3 \qquad (52)$$

where $\phi^0, \theta^0, \psi^0, t_1^0, t_2^0$, and t_3^0 are some approximations, and $\Delta\phi, \Delta\theta, \Delta\psi, \Delta t_1, \Delta t_2$, and Δt_3 are their corresponding corrections. We assume that the corrections Δ's are small and the collinearity equations are linear over the small intervals between the true values of these parameters and their corresponding approximation.

Let

$$F_{n1} = u_{n1} - f\frac{r_1 y_n + t_1}{r_3 y_n + t_3} \qquad (53)$$

$$F_{n2} = u_{n2} - f\frac{r_2 y_n + t_2}{r_3 y_n + t_3}. \qquad (54)$$

These equations can be linearized by Newton's first order

approximation as follows:

$$F_{n1} \simeq F_{n1}^0 + v_{n1} + b_{n11}\Delta\phi + b_{n12}\Delta\theta + b_{n13}\Delta\psi$$
$$+ b_{n14}\Delta t_1 + b_{n15}\Delta t_2 + b_{n16}\Delta t_3$$

$$F_{n2} \simeq F_{n2}^0 + v_{n2} + b_{n21}\Delta\phi + b_{n22}\Delta\theta + b_{n23}\Delta\psi$$
$$+ b_{n24}\Delta t_1 + b_{n25}\Delta t_2 + b_{n26}\Delta t_3$$

where

$$b_{ni1} = \left(\frac{\partial F_{ni}}{\partial\phi}\right)^0 \qquad b_{ni2} = \left(\frac{\partial F_{ni}}{\partial\theta}\right)^0$$

$$b_{ni3} = \left(\frac{\partial F_{ni}}{\partial\psi}\right)^0 \qquad b_{ni4} = \left(\frac{\partial F_{ni}}{\partial t_1}\right)^0$$

$$b_{ni5} = \left(\frac{\partial F_{ni}}{\partial t_2}\right)^0 \qquad b_{ni6} = \left(\frac{\partial F_{ni}}{\partial t_3}\right)^0$$

for $i = 1,2$, where the superscript 0 implies that the function values are computed with the approximations $(\phi^0, \theta^0, \psi^0, t_1^0, t_2^0, t_3^0)$. Taking $F_{n1} = F_{n2} = 0$, the linearized equation can be expressed as the matrix system

$$\begin{pmatrix} b_{111} & b_{112} & b_{113} & b_{114} & b_{115} & b_{116} \\ b_{121} & b_{122} & b_{123} & b_{124} & b_{125} & b_{126} \\ & & \cdots & & & \\ b_{N11} & b_{N12} & b_{N13} & b_{N14} & b_{N15} & b_{N16} \\ b_{N21} & b_{N22} & b_{N23} & b_{N24} & b_{N25} & b_{N26} \end{pmatrix} \begin{pmatrix} \Delta\phi \\ \Delta\theta \\ \Delta\psi \\ \Delta t_1 \\ \Delta t_2 \\ \Delta t_3 \end{pmatrix}$$

$$= \begin{pmatrix} -F_{11}^0 \\ -F_{12}^0 \\ \vdots \\ -F_{N1}^0 \\ -F_{N2}^0 \end{pmatrix} - \begin{pmatrix} v_{11} \\ v_{12} \\ \vdots \\ v_{N1} \\ v_{N2} \end{pmatrix}$$

or simply

$$B\Delta = F - v. \qquad (55)$$

This equation can be solved using the singular value decomposition method. The computed corrections $\Delta = (\Delta\phi, \Delta\theta, \Delta\psi, \Delta t_1, \Delta t_2, \Delta t_3)'$ from one iteration are used to update the parameters $\Lambda = (\phi^0, \theta^0, \psi^0, t_1^0, t_2^0, t_3^0)'$ and then these updated parameters are used as approximations in the next iteration. The whole iteration process is repeated until the corrections become negligibly small.

C. Robust M-Estimation

This section repeats some robust techniques used in nonlinear regression problems as mentioned in Section II. In particular it can be used to solve robustly the equation

$B\Delta = F - v$ that results from the linearization of the original pose estimation problem. Any estimate T_k defined by a minimization problem of the form

$$\min_{T_k} \sum_{i=1}^{n} \rho(x_i - T_k) \qquad (56)$$

or by an implicit equation

$$\sum_{i=0}^{n} \psi(x_i - T_k) = 0 \qquad (57)$$

where ρ is an arbitrary function (called object function),

$$\psi(x - T_k) = \frac{\partial}{\partial T_k} \rho(x - T_k) \qquad (58)$$

is called an M-estimate. This last equation can be written equivalently as

$$\sum_{i=0}^{n} w_i(x_i - T_k) = 0 \qquad (59)$$

where

$$w_i = \frac{\psi(x_i - T_k)}{x_i - T_k}, \qquad i = 1, \cdots, N. \qquad (60)$$

This gives a formal representation of T_k as a weighted mean

$$T_k = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} \qquad (61)$$

with weights depending on the sample (Huber [9]). It is known that M-estimators minimize objective functions more general than the familiar sum of squared residuals associated with the sample mean. Among many forms of functions ρ and ψ proposed in the literature, Huber's and Tukey's form is investigated in this experiment. Huber derived the following robust ρ and ψ.

$$\rho(x) = \begin{cases} 0.5x^2, & \text{if } |x| \le a; \\ a|x| - 0.5a^2, & \text{otherwise.} \end{cases}$$

$$\psi(x) = \begin{cases} -a, & \text{if } x < -a; \\ x, & \text{if } |x| \le a; \\ a, & \text{if } x > a. \end{cases}$$

Tukey's ψ function can be expressed as

$$\psi(x) = \begin{cases} x(1 - (x/a)^2)^2, & \text{if } |x| \le a; \\ 0, & \text{if } |x| > a. \end{cases} \qquad (62)$$

where a is a tuning constant, 1.5 for Huber's and 6 for Tukey's.

The nonlinear regression problem can be formulated as follows. Let $f_i: E^m \rightarrow E$, $i = 1, \cdots, n$ be functions that map m-dimensional space into a real line. Let $\theta = (\theta_1, \theta_2, \cdots, \theta_m)' \in E^m$ be the m-dimensional unknown vector to be estimated. The solution to the set of n equations

$$f_i(\theta) = y_i, \qquad i = 1, \cdots, n \qquad (63)$$

that minimizes

$$\sum_{i=1}^{n} \rho\left(\frac{y_i - f_i(\theta)}{S}\right) \qquad (64)$$

can be found in several different ways. To create a scale invariant version of the M-estimator, the robust estimate of scale such as the following is introduced.

$$S = \frac{\text{median}_i |y_i - f_i(\theta)|}{0.6745} \qquad (65)$$

where 0.6745 is one half of the interquantile range of the Gaussian normal distribution $N(0,1)$. Here we take the median of the nonzero deviations only because, with large m, too many residuals can equal zero (Hogg [8]).

In robust estimation, the estimates are obtained only after an iterative process because the estimates do not have closed forms. Two such iterative methods are presented here that can solve the minimization problem stated previously (Huber [9]).

1) Modified Residual Method: In this method the residuals are modified by a proper ψ function before the least-squares problem is solved. The iterative procedure to determine θ is as follows.

1) Choose an initial approximation θ^0.
2) Iterate. Given the estimation θ^k in step k, compute the solution in the $(k+1)$th step as follows.
   a) Compute the modified residuals r_i^* for $i = 1, \cdots, n$ as

$$r_i^* = \psi\left(\frac{r_i}{S^k}\right) S^k$$

where

$$r_i = y_i - f_i(\theta^k)$$

$$S^k = \underset{r_i \ne 0}{\text{median}} |r_i| / 0.6745.$$

   b) Solve the least-squares problem $X\delta = r^*$, where $X = [x_{ij}]$ is the gradient matrix as

$$x_{ij} = \frac{\partial}{\partial \theta_j} f_i(\theta^k).$$

   The solution for this equation can be found using the standard least-squares method. If the singular value decomposition of the matrix X is $X = U\Sigma V'$, then the solution is $\hat{\delta} = V\Sigma^{-1}U'r^*$.
   c) Set $\theta^{k+1} = \theta^k + \hat{\delta}$.

2) Modified Weights Method: Taking the derivative of the objective function ρ with respect to θ and set it to zero, we get

$$\sum_i \psi\left(\frac{y_i - f_i(\theta)}{S}\right) \frac{\partial f_i(\theta)}{\partial \theta_j} = 0. \qquad (66)$$

In the standard weighted form

$$\sum_i w_i r_i \frac{\partial f_i(\theta)}{\theta_j} = 0 \qquad (67)$$

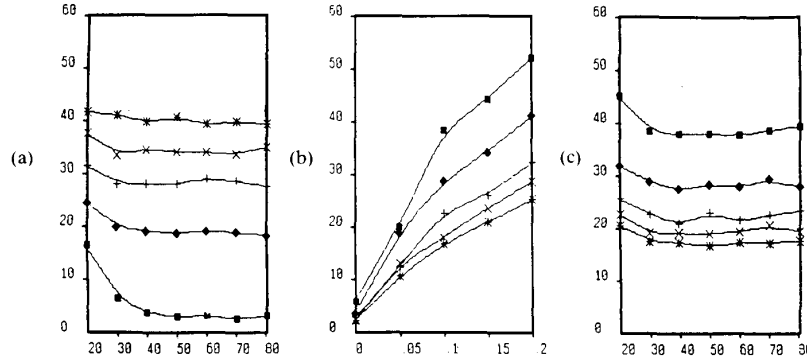Fig. 9.   Illustrates performance characteristics for initial approximation solution (Method 2). (a) SNR = 20 dB. Outliers in percent are ■ = 0, ◆ = 5, + = 100, × = 15, * = 20. (b) SNR = 40 dB. ■, $N = 10$; ◆, $N = 20$; +, $N = 30$; ×, $N = 40$; *, $N = 50$. (c) Outlier = 10 percent. ■, $N = 10$; ◆, $N = 20$; +, $N = 30$; ×, $N = 40$; *, $N = 50$.

where

$$w_i = \frac{\psi\left(\dfrac{r_i}{S}\right)}{\left(\dfrac{r_i}{S}\right)}. \tag{68}$$

Therefore the iterative procedure to determine θ is as follows.

1) Choose an initial approximation θ^0.
2) Iterate. Given θ^k at kth step, compute θ^{k+1} as follows.
   a) Solve

$$PX\delta = Pr$$

where

$$P = \begin{pmatrix} \sqrt{w_1} & & \\ & \ddots & \\ & & \sqrt{w_N} \end{pmatrix}.$$

   b) If $\hat{\delta}$ is the solution in Step 2a), then set

$$\theta^{k+1} = \theta^k + \hat{\delta}.$$

D. Experimental Results

To measure the performance of the pose estimation algorithms, several hundred thousand controlled experiments were performed. This section describes how the controlled experiments are constructed and shows the results from those experiments. The result is presented as a graph where the sum of errors of the three rotation angles, ϕ, θ, ψ, is plotted against various control parameters such as the SNR, the number of matched points, or the number of outliers, which will be defined later.

1) Data Set Generation: A set of 3-D model points, $y_i = (y_{i1}, y_{i2}, y_{i3})'$, $i = 1, \cdots, N$, are generated within a box defined by

$$y_{i1}, y_{i2}, y_{i3} \in [0, 10].$$

That is the three coordinates are independent random variables each of them uniformly distributed between 0 and 10. Next three rotation angles are selected from an interval [20,70] and the translation vector $t = (t_1, t_2, t_3)$ is also generated such that t_1 and t_2 are uniformly distributed within an interval [5,15] and t_3 is within [20,50]. Having these transformation parameters, the 3-D model points are rotated and translated in the 3-D space forming a set of 3-D points x_i, $i = 1, \cdots, N$. At this stage independent identically distributed Gaussian noise $N(0, \sigma)$ is added to all three coordinates of the transformed points x_i. To test the robustness of the algorithms, some fraction of the 3-D points, x_i, are replaced with randomly generated 3-D points, $z_i = (z_{i1}, z_{i2}, z_{i3})'$, $i = 1, \cdots, M$. M is the number of the replaced 3-D points and

$$z_{i1} = t_1 + \nu_{i1}$$

$$z_{i2} = t_2 + \nu_{i2}$$

$$z_{i3} = x_{i3} \tag{69}$$

where ν_{i1}, ν_{i2}, $i = 1, \cdots, M$ are independent random variables uniformly distributed within an interval $[-5,5]$. These random points, z_i, are called outliers in our experiments. To get the matching set of 2-D points, x_i, $i = 1, \cdots, N$ are perspectively projected onto the image plane. Given the 3-D model points and the corresponding 2-D points on the image plane, each algorithm is applied to find the three rotation angles and the translation vector.

One can notice from the above description that there are three parameters we can control in each experiment. They are the number of 3-D model points N, the standard deviation σ of the Gaussian noise, and the number of outliers M. In the experimental result, we use SNR and the percent of outliers PO, in place of σ and M respectively, where

$$SNR = 20 \log \frac{10}{\sigma} \text{ dB}, \tag{70}$$

$$PO = \frac{M}{N} \times 100 \text{ percent}. \tag{71}$$

2) Results: For each parameter setting (N, SNR, PO) 1000 experiments are performed to get a reasonable estimate of the performance of the algorithms. For each
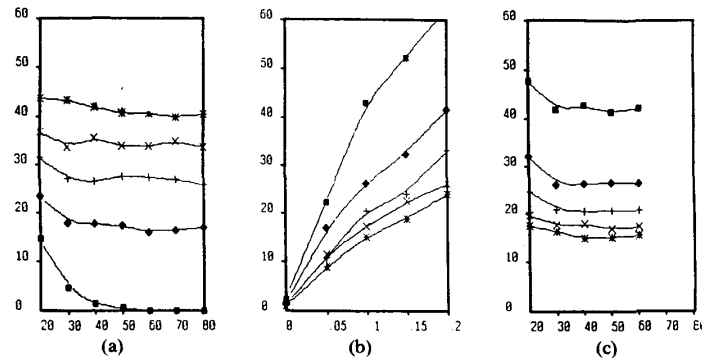
Fig. 10.   Illustrates performance characteristics of least-squares adjust by linearization. Legend same as in Fig. 9.
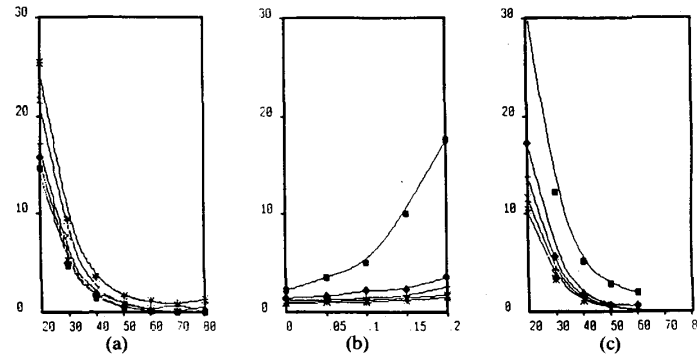

Fig. 11.   Illustrates performance characteristics of robust M-estimate algorithm. Legend is same as in Fig. 9.

algorithm we performed three different sets of experiments (E1, E2, and E3), as follows.

E1:   Set N = 20. Estimate the sum of three rotation angle error against SNR (20 dB–80 dB in 10 dB step) for different PO (0 percent to 20 percent in 5 percent step).

E2:   Set SNR = 40 dB. Estimate the sum of three rotation angle error against PO (0 percent–20 percent in 5 percent step) for different N (10 to 50 by steps of 10).

E3:   Set PO = 10 percent. Estimate the sum of three rotation angle error against SNR (20 dB–80 dB in 10dB step) for different N (10 to 50 by steps of 10).

Fig. 9 shows the results of E1, E2, and E3 performed for the initial approximation algorithm using iterative least-squares solution (A1), Method 2 of Section IV-A-1. Initial estimate for the approximate distance is set to 10 in all experiments. For the linearized algorithms, the initial estimate of the three rotation angles are selected randomly within 15 degrees of the true angles. The initial approximate of the translation vector is selected randomly within ±10 of the true translation vector. Figs. 10 and 11 show the result of the least-squares adjustment by linearization algorithm (A2), algorithm in Section IV-A-1, and the robust M-estimate algorithm (A3), modified weights algo-

rithm in Section IV-A-2, respectively. Fig. 12 compares the three algorithms A1, A2, and A3 in the experiment set E1. Figs. 13 and 14 compare the three algorithms in the experiment set E2 and E3 respectively. One more experiment is performed to compare the algorithms A2 and A3. With N = 20 and PO = 10 percent, algorithms A2 and A3 are applied for SNR from 20 dB–40 dB in a step of 10 dB, and the algorithm A2 is applied for N = 18, PO = 0 percent and SNR from 20 dB to 40 dB in a step of 10 dB. This compares the efficiency of the robust technique against the nonrobust technique in the case where the nonrobust technique uses only the nonoutlier points given to the robust technique. Fig. 15 shows the result of this experiment.

V.   2-D PERSPECTIVE–2-D PERSPECTIVE PROJECTION POSE ESTIMATION

The estimation of three-dimensional motion parameters of a rigid body is an important problem in motion analysis. Its applications include scene analysis, motion prediction, robotic vision, and on line dynamic industrial processing. There has been much literature contributed to 3-D parameter estimation, but few of these contributions systematically discuss the effect of noise. Thompson [19] developed the nonlinear equations using the form resulting from the correspondence of 2-D perspective projection points on one image with 2-D perspective projection points
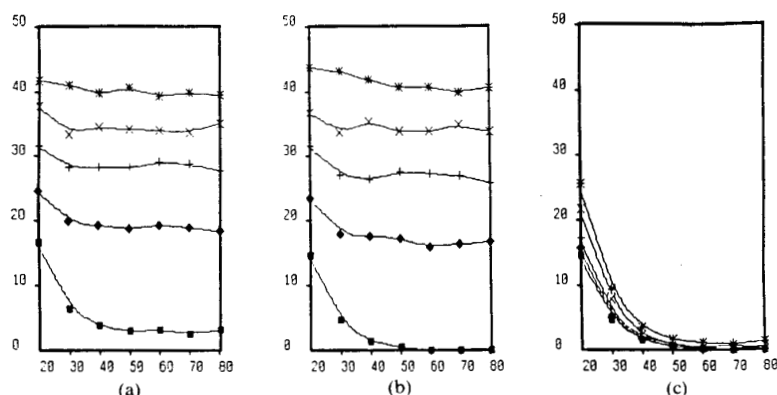
Fig. 12. Illustrates performance characteristics of angle error as function of SNR for initial approximation method, nonrobust linearized least-squares adjustment, and robust *M*-estimate. Legend same as in Fig. 9. Outlier, SNR = 40 dB. ■ = 10; ◆ = 20; + = 30; × = 40; ✳ = 50.
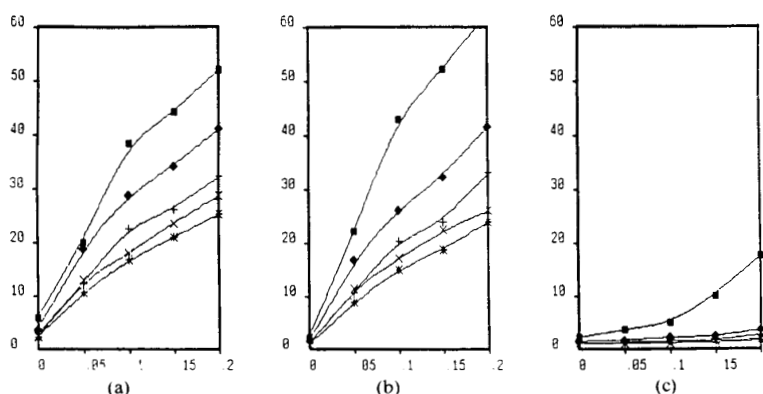

Fig. 13. Illustrates performance characteristics of angle error versus fraction of outliers for initial approximation method, linearized least-squares adjustment, and robust *M*-estimate. Refer to Fig. 12 for legend.
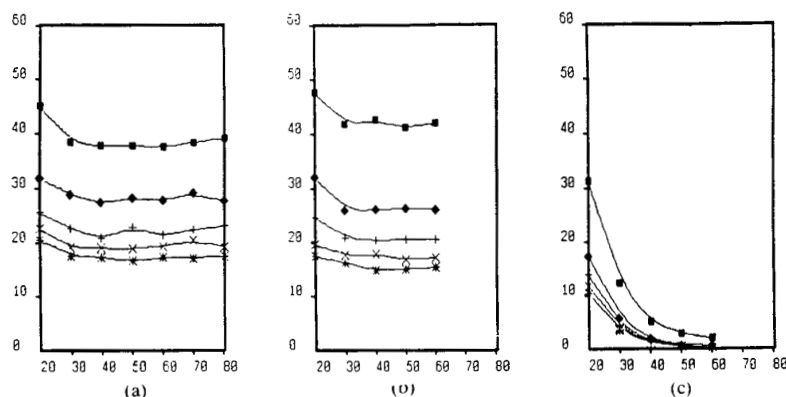

Fig. 14. Illustrates performance characteristics of angle error versus fraction of outliers for initial approximation method, linearized least-squares adjustment, and robust *M*-estimate. Outlier is 10 percent. Refer to Fig. 12 for legend.

on another image. He gave a solution that determines a rotation matrix guaranteed to orthonormal. His method was to linearize the nonlinear equations and iterate. Roach and Aggarwal [12] developed a nonlinear algorithm and dealt with noisy data. Their results show that accuracy can be improved by increasing the number of corresponding point pairs; but the number of corresponding point pairs in their experiments is too few (15 corresponding point

pairs). The linear motion parameters estimation algorithm was developed by Longuet–Higgins [10], extended by Tsai and Huang [21], unified by X. Zhuang, T. S. Huang, and R. M. Haralick [23], and simplified by X. Zhuang and R. M. Haralick. The linear algorithm has an advantage of being simple and fast over the nonlinear algorithm. Furthermore it can always find a unique solution except in degenerate cases. The linear algorithm works very well
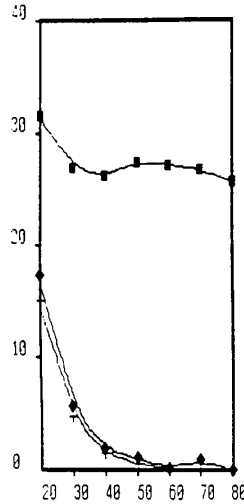
Fig. 15. Illustrates efficiency of robust technique operating on data set of 20 points, 18 points having Gaussian noise and 2 outliers, against nonrobust technique operating on data set having 18 points having Gaussian noise. ■ = nonrobust (20), ◆ = robust (20), + = nonrobust (18). Outlier = 10 percent.

when there is limited noise and no corresponding point matching errors. However the algorithm is highly sensitive to noise and matching errors. Experiments show that when combined with real world image corresponding point data produced by a vision systems, a disaster occurs. Increasing the number of corresponding point pairs can to some extent suppress the noise effect. The main problem in linear algorithm is the least-squares estimation.

The method of least-squares is based on an evaluation of the magnitude of residuals and is sensitive to gross errors, matching errors and outliers. Unlike the least-squares estimator the robust estimator has good resistance and robustness to gross matching error and outliers. In this section a simplified linear algorithm presented by Zhuang [24] is used to get the baseline noise behavior of the linear algorithm. The principle of robust computation is presented. The experimental design is discussed and the results shows that robust algorithm has better performance and stability.

A. Simplified Linear Algorithm

As shown in Fig. 16, we assume that the coordinate system is the camera reference frame, the origin being the center of the lens. A rigid body is in motion in the half-space $z \leqslant 0$. Let $P = (x, y, z)^t$ represent the object point coordinates before motion and $P' = (x', y', z')^t$ represent the same object point coordinates after motion. Let (X, Y), (X', Y') represent the perspective coordinate of P and P' onto the image plane $z = 1$. These give

$$X = x/z$$
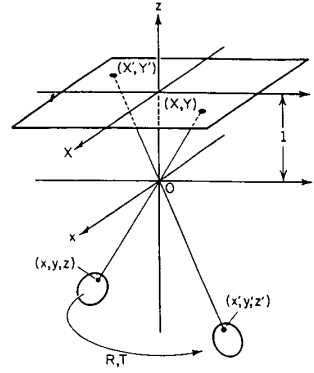$$Y = y/z$$
$$X' = x'/z'$$
$$Y' = y'/z'. \tag{72}$$

Fig. 16. Imaging Geometry.

The rigid body motion equation is given as

$$P' = R_o P + T_o \tag{73}$$

where R_o is an 3×3 rotation matrix (orthonormal); T_o is 3×1 translation vector.

The problem is to estimate rotation matrix R_o and translation matrix T_o.

1) The Two View Motion Equation: Choosing any nonzero vector T which is collinear with T_o and taking its cross-product with both sides of (73) we obtaiin

$$\frac{z'}{z} T \times (X', Y', 1)^t = T \times \left[R_o (X, Y, 1)^t \right]. \tag{74}$$

Taking inner product of both sides of (74) with $(X', Y', 1)$ yields

$$(X', Y', 1)(T \times R_o)(X, Y, 1)^t = 0 \tag{75}$$

where $T \times R_o = [T \times r_1, T \times r_2, T \times r_3]$, and r_1, r_2, r_3 are the columns of R_o. Define the motion parameter matrix E by

$$E = T \times R_o. \tag{76}$$

For any image corresponding pair $[(X, Y), (X', Y')]$ the matrix E satisfies the following linear homogeneous equations with respect to (wrt) nine elements of E

$$(X', Y', 1) E (X, Y, 1)^t = 0. \tag{77}$$

Relation (77) was originally shown by Thompson [19]. Suppose that we have N corespondences. Then E can be estimated from the following equation. Let

$$A = \begin{pmatrix} X_1'X_1 & X_1'Y_1 & X_1' & Y_1'X_1 & Y_1'Y_1 & Y_1' & X_1 & Y_1 & 1 \\ X_2'X_2 & X_2'Y_2 & X_2' & Y_2'X_2 & Y_2'Y_2 & Y_2' & X_2 & Y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_n'X_n & X_n'Y_n & X_n' & Y_n'X_n & Y_n'Y_n & Y_n' & X_n & Y_n & 1 \end{pmatrix}$$

$$E = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix}$$

$$h = (h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9)^t.$$

Then (77) can be transformed into the overconstraint linear equation for h

$$Ah = 0. \tag{78}$$

Solving (78) in the least-squares sense we seek an estimator h that minimizes $\|Ah\|^2$. The nine component vector h is found to be the right eigenvector of A having smallest singular value. Any $T \times R_o$ with $T \times T_o = 0$ satisfies (77). Moreover such a colinear vector T has one degree of freedom when $T_o \neq 0$ or three degrees of freedom when $T_o = 0$. Thus the general solution of the two-view motion (77) has at least one degree of freedom when $T_o \neq 0$ or three degrees of freedom when $T_o = 0$.

When $T_o \neq 0$, the nine elements of E must have a rank 8, and $T_o = 0$ the nine elements of E must have rank 6. Under the surface assumption (Zhuang, Haralick, and Huang, [23]) the number of image corresponding point pairs must be at least 8 when $T_o \neq 0$, or greater than or equal to 6 when $T_o = 0$. The geometry interpretation we use assumes that the object is stationary and the camera is moving. Let the origin of the camera system be O and O' respectively before and after motion. Then the surface assumption holds if and only if the 3-D points corresponding to the observed image points do not lie on a quadratic surface passing through O and O' when $T_o \neq 0$ or a cone with its apex at O when $T_o = 0$.

2) Decomposing E: E has two decompositions; $T \times R_o$ and $(-T) \times R_o$ with R_o being an orthonormal matrix of the first kind. In order to determine the correct decomposition we note that $E = [T \times r1, T \times r2, T \times r3]$. Hence its three columns span a 2-D space and also $\|E\| = \sqrt{2}\|T\|$. Therefore we can get three constraints as follows:

$$\text{rank}(E) = 2$$

$$\|E\| = 2\|T\|$$

$$E^t T = 0. \tag{79}$$

We can use the least-squares method to solve (79) for T and obtain the value of the T vector from the other two constraints. Since T is colinear with T, T should have the same orientation as T or $-T$. Taking a cross-product with both sides of (73) by $(X', Y', 1)^t$ we obtain

$$z(X', Y', 1)^t \times \left[R(X, Y, 1)^t \right] + (X', Y', 1)^t \times T = 0. \tag{80}$$

Since $z < 0$, it implies that T_0 has the same orientation as T or $(-T)$ if and only if $(X', Y', 1)^t \times [R_0(X, Y, 1)^t]$ has the same orientation as $(X', Y', 1)^t \times T$ or $[-(X', Y', 1)^t \times T]$. This implies that it has the same orientation if and only if

$$\sum_{i=1}^{n} (X_i', Y_i', 1)^t \times \left[R_0(X_i, Y_i, 1)^t \right] (X_i', Y_i', 1)^t \times T$$

$$\geq 0 \text{ or } \leq 0. \tag{81}$$

Once the correct T is determined, the true R_0 could be uniquely determined through $E = T \times R_0$ as follows:

$$R_0 = [E_2 \times E_3, E_3 \times E_1, E_1 \times E_2] - T \times E \tag{82}$$

where

$$E = [E_1, E_2, E_3].$$

B. The Robust Algorithm

As mentioned in the previous section (78) can be solved by least-squares estimator. However it is sensitive to gross errors. In this section the robust algorithm is presented. The robust algorithm is an iterative reweighted least-squares estimation procedure where the weights are recomputed each iteration and are computed as a biweight. The difference between the biweight estimator and the least-squares estimator is briefly discussed.

1) Biweight Estimator: Let x_i be the ith observation and $\hat{x}$ be estimated mean value of the observations. The least-squares method minimizes the residual error

$$\epsilon^2 = \sum_{i=1}^{n} (x_i - \hat{x})^2$$

and the object function, ρ, is expressed as follows

$$\rho(x_i; \hat{x}) = (x_i - \hat{x})^2. \tag{83}$$

To find the solution of problem we differentiate ρ wrt $\hat{x}$. The derivative ψ satisfies

$$\sum_{i=1}^{n} \psi(x_i; \hat{x}) = \sum_{i=1}^{n} (x_i - \hat{x}) = 0. \tag{84}$$

As discussed in Hoaglin the least-squares estimator is linear and unbound.

The ψ function of the biweight estimator can be represented as follows:

$$\psi(u) = \begin{cases} u(1 - u^2)^2, & |u| \leq 1, \\ 0, & \text{otherwise,} \end{cases} \tag{85}$$

where

$$u_i = \frac{f_i(\epsilon)}{cs_n}$$

$f_i(\epsilon)$: residual error function
s_n: median value of $f_i(\epsilon)$
c: tuning constant.

Unlike the least-squares estimator, the ψ-function of the biweight estimator is bounded. When the value of tuning constant is small it will delete a lot of useful data. On the other hand, when the value is large the outliers cannot be removed from the images. Hence the tuning constant depends on the value of gross errors. A reasonable value range for tuning constant is from 4 to 12. In here we let $c = 4$. Let $\psi(u) = w(u)u$. Thus the weight function $w(u)$ can be represented by

$$w(u) = \begin{cases} [1 - u^2]^2, & \text{if } |u| \leq 1; \\ 0, & \text{otherwise.} \end{cases} \tag{86}$$

2) Robust Estimation of E: From the previous equation we can see that the biweight estimator is a weighted least-square estimator. With the weight matrix we rewrite (78) as
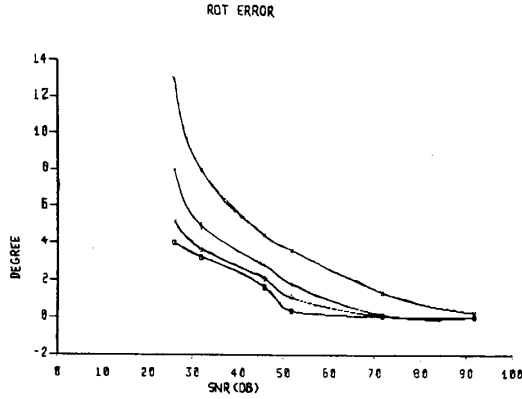
$$WAh = 0. \tag{87}$$

Fig. 17. Mean error between estimated rotation angles and true rotation angles versus Gaussian noise level for four corresponding point data set sizes of 8–110 pairs. Each point on the graph represents 1000 trials. — ■ $N = 110$, — ◇ $N = 50$, — ⏐ $N = 20$, — ✕ $N = 0$.

To find the value of h that minimizes $\|WAh\|^2$ the singular value decomposition can be used

$$WA = U\sum V^t \qquad (88)$$

where

$$\sum_{m \times n} = \begin{pmatrix} s_1 & 0 & \cdots & 0 \\ 0 & s_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & s_9 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 \end{pmatrix}$$

$$V^t_{n \times n} = [v_1, v_2, \cdots, v_n]$$

$$U_{m \times m} = [u_1, u_2, \cdots, u_m].$$

The index n is 9 and m is the number of corresponding point pairs. The eigenvector of V that corresponds to the smallest nonzero eigenvalue in $\sum$ is the solution of weighted least squares. Here it will be denoted by v_9. Multiplying the current solution for h by A to get the new residual. Gross errors are not necessarily accompanied large residuals as explained in Huber [9]. Hence the residual errors need to be adjusted according to the following

$$f_i(\epsilon) = \frac{\epsilon_i}{1 - h_{ii}}. \qquad (89)$$

where h_{ii} is the diagonal element of the projection matrix H

$$H = (WA)\big((WA)^t(WA)\big)^{-1}(WA)^t. \qquad (90)$$

We can simplify this equation by substituting $U\sum V^t$ for WA. After some linear algebra manipulation (90) becomes

$$H = U_\alpha U_\alpha^t \qquad (91)$$

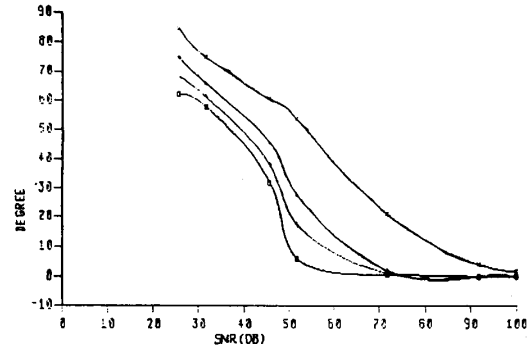where $U_{am \times 9} = [u_1, u_2, \cdots, u_9]$.

Fig. 18. Mean angle error between calculated translation vector and true translation vector versus Gaussian noise level for four corresponding point data set sizes of 8–110 pairs. Each point on graph represents 1000 trials. Legend same as in Fig. 17.

TABLE I
SNR (dB) FOR MEAN ABSOLUTE ERROR IN 1 DEGREE

	Rotation Angles				Translation Vector			
No. of point pairs	8	20	50	110	8	20	50	110
Gaussian	75	57	52	50	105	78	73	68
Uniform	74	56	52	49	106	78	72	68

It is trivial then to obtain

$$h_{ii} = \sum_{k=1}^{9} u_{ik}^2.$$

Once h_{ii} are obtained, then they can be substituted into (89) to get the new residual error function and to update the weight matrix. The initial weight matrix is identity matrix. The iterations continue until some criteria are satisfied. In our experiments when the error ϵ^2 is less than $0.001\epsilon^2$ of first iteration or the iteration number is larger than 25, then the iteration process stops. Usually it will converge after a few iterations. The value of v_9 at the last iteration is the robust fitting solution.

C. Simulation Result and Discussion

In this section we discuss the experimental results of a large number of controlled experiments using the linear algorithm and the robust algorithm under a varying amount of noise, gross errors and corresponding point pairs. As shown in Fig. 16, the image frame is located at $z = 1$. By mapping 3-D spatial coordinates into image frame, and then adding noise to the points before and after motion, we obtain

$$\begin{pmatrix} X(t) \\ Y(t) \end{pmatrix} = \begin{pmatrix} 1/z(t) & 0 & 0 \\ 0 & 1/z(t) & 0 \end{pmatrix} \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} + \begin{pmatrix} n_x(t) \\ n_y(t) \end{pmatrix}. \qquad (92)$$

Signal is related to object image size, and noise may come from camera error, digitization, or corresponding point extraction error. Define SNR $= 20 \log(\text{signal}/\sigma)$ dB, where σ is the standard deviation. In the simulation experiments, the 3-D spatial coordinates before motion (x, y, z), true
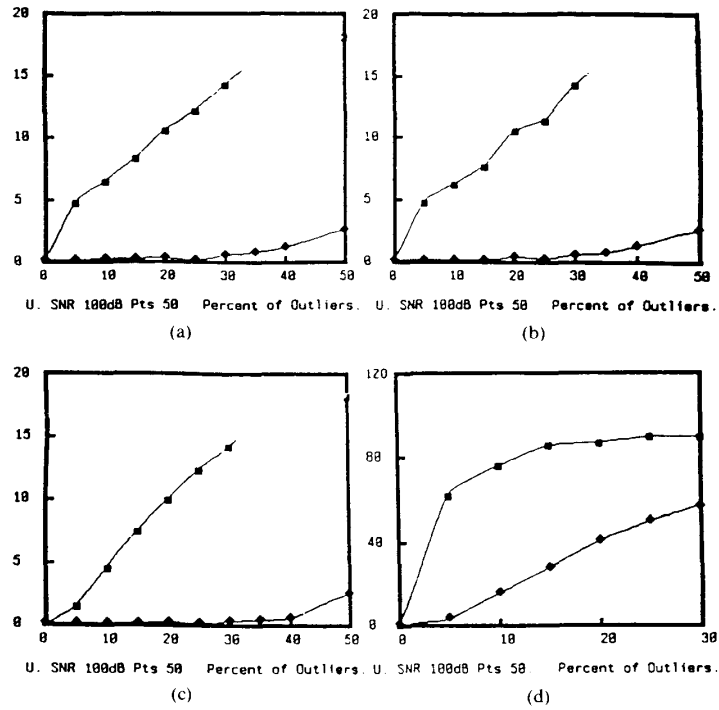
Fig. 19. Compares ϕ, ψ, θ angle error and translation angle error between linear algorithm and robust algorithm for different percent of outliers. Noise is uniform with 100 dB SNR. The number of points is 50. Each point on the graph represents 1000 trials. (a) Rotation angle ϕ error mean. (b) Rotation angle ψ error mean. (c) Rotation angle θ error mean. (d) Translation vector error mean. ■ is linear algorithm, ◆ is robust algorithm.

Fig. 20. Compares ϕ, ψ, θ angle error and translation angle error between linear algorithm and robust algorithm for different number of points. Noise is uniform with 100 dB SNR. The percent of outliers is 10 percent. Each point on the graph represents 1000 trials. See Fig. 19 for legend.
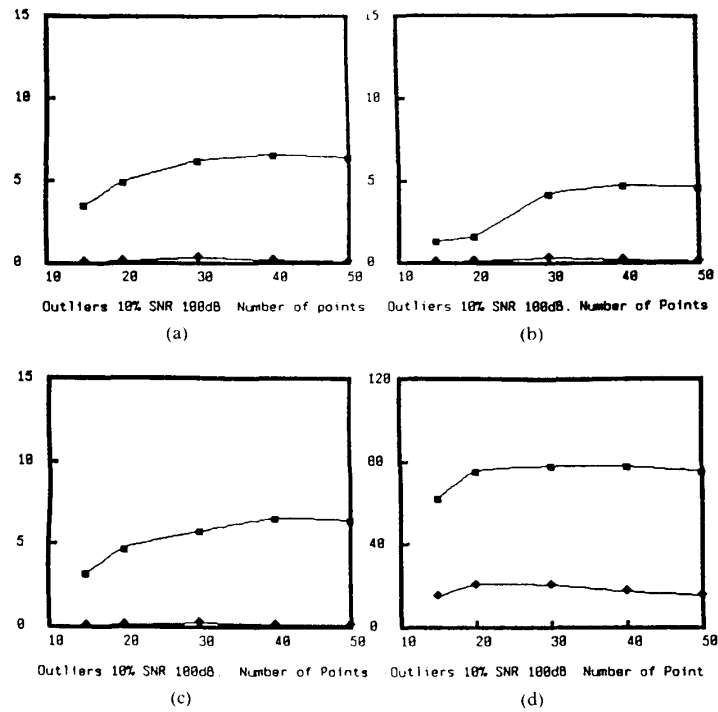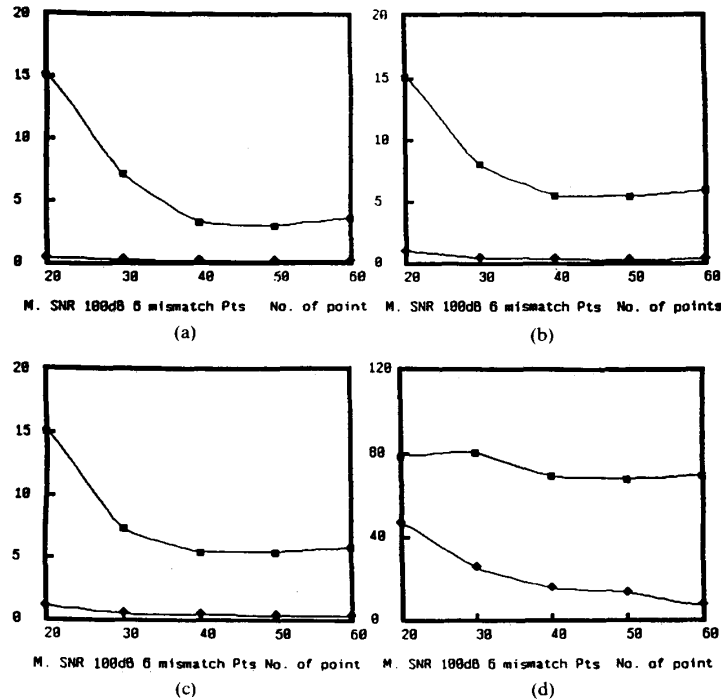
Fig. 21. Compares ϕ, ψ, θ angle error and translation angle error between linear algorithm and robust algorithm for different number of points. Noise is uniform with 100 dB SNR and is added six points of mismatch. Each point on the graph represents 1000 trials. (a) Rotation angle ψ error mean. (b) Rotation angle ϕ error mean. (c) Rotation angle θ error mean. (d) Translation vector error mean.

matrix R_o, and true translation vector T_o are generated by a random number generator. The 3-D data are generated within the $(-2, -2, -2)$ to $(2,2,2)$ cube. The rotation angles ϕ, θ, ψ are generated within the range of $[-15,15]$ degree and translation vectors are chosen within the range $(-0.5, -0.5, -0.5)$ to $(0.5,0.5,0.5)$ cube. Then the 3-D spatial coordinates after motion (x', y', z') can be calculated in the natural way. Projecting the 3-D spatial coordinates into the image frame we get perspective coordinates. Noisy image data is obtained by adding Gaussian noise with zero mean to the image coordinates. Outliers are generated by randomly moving some corresponding points position in image frame after motion. The number of outliers are chosen as a percent of corresponding point pairs. Following the linear algorithm or the robust algorithm as described above we can get the calculated rotation matrix and translation vector. From the calculated rotation matrix the calculated ϕ, θ, ψ are obtained. We compare the difference between the calculated ϕ, θ, ψ and the true ϕ, θ, ψ in terms of mean absolute error. For each experimental condition a thousand trials are done. Mismatching noise is simulated by randomly swapping one component from a pair of corresponding points. The percent of mismatch is the ratio of mismatching points to number of corresponding points.

The number of corresponding point pairs varies from the 8-point pairs to 110-point pairs in four steps. When noise-free, the linear algorithm has excellent performance with zero error for all cases. Figs. 17 and 18 show the translation error and rotation degree error, which can define an average of mean absolute error of three Euler angles, versus the SNR for different numbers of corresponding point pairs for Gaussian noise. It shows that the error increases as the noise level increases. Furthermore depending on the number of corresponding point pairs, the error increases very rapidly when the SNR gets below a knee value. Table I shows the minimum SNR to guarantee a less than 1 degree error as a function of numbers of corresponding point pairs and kind of noise distribution.

The robust experiments show that the robust estimators can protect from outliers almost up to a fraction of 50 percent. The linear algorithm breaks down when only a small percent of outliers is present. Similar results occur in the mismatch experiments. Figs. 19(a)–(d) show the effect of outliers to both the linear and robust algorithm. The error of the linear algorithm almost increases linearly, but the robust algorithm shows much better performance and stability. The error of ψ is approximately twice less than the error for θ and ϕ. The azimuth and tilt angle are more vulnerable to noise than swing angle. In Figs. 20(a)–(d) we fix the percent of outliers and increase the number of corresponding points. Because the outlier percentage is constant, the mean error is approximately constant as the number of corresponding points increase. The mismatch error results are shown in Figs. 21(a)–(d). They show results similar to the outlier results.

D. Summary of Robust Algorithm

Step 0) Use the identity matrix for initial weight matrix.
Step 1) Use singular value decomposition to solve (87).
Step 2) Update the weight matrix by (86).
    Repeat Steps 1) and 2) until the criteria is satisfied.
Step 3) Determine the translation vector from (79) and (81).
Step 4) Obtain true R_o from (82).

VI. CONCLUSION

We have presented solutions to four pose estimation problems and have characterized the performance of these algorithms in simulation experiments with the noise model being additive Gaussian noise, uniform noise, outliers noise, or mismatch noise. We have observed in these experiments a knee phenomenon. When the signal to noise ratio gets to be below a knee, the RMS error skyrockets. When the number of corresponding point pairs gets to be below a knee value, the RMS error also skyrockets. The iterative weighted least-squares technique is proved robust to the blunder data.

REFERENCES

[1] J. A. R. Blais, "Three-dimensional similarity," *The Canadian Surveyor*, no. 1, 1972, pp. 71–76.
[2] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, no. 5, pp. 698–700, Sept. 1987.
[3] J. Q. Fang, and T. S. Huang "Some experimentson estimating the 3-D motion parameters of a rigid body from two consecutive image frames," *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-6, pp. 547–554, 1984.
[4] M. A. Fischler and R. S. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, June 1981.
[5] W. Förstner, The reliability of block triangulation, *Photogrammetric Engineering and Remote Sensing*, vol. 51, no. 6, pp. 1137–1149, Aug. 1985.
[6] S. I. Granshaw, "Relative Orientation Problems," *Photogrammetric Record*, vol. 9, no. 53, pp. 669–674, 1979.
[7] Hoaglin, C. F. Mosteller, and J. W. Tukey, *Understanding Robust and Exploratory Data Analysis*. New York: John Wiley, 1983, pp. 348–349.
[8] Robert J. Hogg, *An Introduction to Robust Estimation, Robustness in Statistics*, R. L. Launer and G. N. Wilkinson, Eds. New York: Academic Press, 1979.
[9] Peter J. Huber, *Robust Statistics*. New York: John Wiley, 1981.
[10] C. A. Longuet-Higgins, "Computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, pp. 133–135, 1981.
[11] J. A. Pope, "An advantageous, alternative parametrization of rotations for analytical photogrammetry," ESSA Tech. Rep., C and GS 39, 1970.
[12] J. W. Roach and J. K. Aggarwal "Determining the movement of objects from a sequence of images," *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-6, pp. 554–562, 1980.
[13] F. Sanso, "An exact solution of the roto-translation problem," *Photogrammetria*, vol. 29, pp. 203–216, 1973.
[14] G. H. Schut, "On exact linear equations for the computation of rotational element of absolute orientation," *Photogrammetria*, vol. 15, no. 1, pp. 34–37, 1960.
[15] Steven A. Shafer and T. Kanade, "Gradient space under orthography and perspective," *Computer Vision Graphics Image Processing*, vol. 24, pp. 182–199, 1983.
[16] ——, "Using shadows in finding surface orientations," Dept. of Computer Science, Carnegie-Mellon Univ., Pittsburgh, PA, pp. 1–61, 1982.
[17] W. Szczepanski, "Die Lösungsvorschläge für den räumlichen Rückwärtseinschnitt, Deutsche Geodätische Kommission," Reihe C: Dissertationen-Heft Nr. 29, pp. 1–144, 1958.
[18] E. H. Thompson, "An exact linear solution of the problem of absolute orientation," *Photogrammetria*, vol. 15, no. 4, pp. 163–178, 1958.
[19] ——, "A rational algebraic formulation of the problem of relative orientation," *The Photogrammetric Record*, vol. 3, no. 14, pp. 152–159, 1959.
[20] J. M. Tienstra, "Calculation of orthogonal matrices," ITC Delft Series, A48, 1969.
[21] R. Y. Tsai and T. S. Huang, "Uniqueness and estimation of 3-D motion parameters of rigid objects with curved surfaces," *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-6, pp. 13–17, 1984.
[22] B. Wrobel and D. Klemm, "Über die Vermeidung singulärer Fälle bei der Berechnung allgemeiner räumlicher Drehungen," in *Proc. XVth Congress of ISPRS*, Rio de Janeiro, Brazil, 1984, and in the *Int. Archives of Photogrammetry and Remote Sensing*, vol. 25, part A3b, pp. 1153–1163.
[23] X. Zhuang, R. Haralick, and T. S. Huang, "Two-view motion analysis: A unified algorithm," *Opt. Soc. Am.*, vol. 3, no. 9, pp. 1492–1450, 1986.
[24] X. Zhuang, "A simplification to linear algorithms to view motion," *Computer Vision Graphics Image Processing*, vol. 46, pp. 175–178, 1989.
[25] R. M. Haralick, H. Joo, C. Lee, X. Zhuang, V. Vaidya, and M. Kim, "Pose estimation from corresponding point data," *IEEE Computer Society Workshop on Computer Vision*, Miami Beach, FL, Nov. 30–Dec. 3, 1987, pp. 258–263.

Robert M. Haralick (S'62–M'69–SM'76–F'84) was born in Brooklyn, NY, on September 30, 1943. He received the B.A. degree in mathematics from the University of Kansas, Lawrence, in 1964, the B.S. degree in electrical engineering in 1966, the M.S. degree in electrical engineering in 1967, and the Ph.D. degree from the University of Kansas in 1969.

He has worked with Autonetics and IBM. In 1965 he worked for the Center for Research, University of Kansas, as a Research Engineer and in 1969 he joined the faculty of the Department of Electrical Engineering there where he last served as a Professor from 1975 to 1978. In 1979 he joined the faculty of the Department of Electrical Engineering at Virginia Polytechnic Institute and State University where he was a Professor and Director of the Spatial Data Analysis Laboratory. From 1984 to 1986 he served as Vice President of Research at Machine Vision International, Ann Arbor, MI. He now holds the Boeing Clairmont Egtvedt chaired professorship in the Department of Electrical Engineering, University of Washington, Seattle. He has done research in pattern recognition, multi-image processing, remote sensing, texture analysis, data compression, clustering, artificial intelligence, and general systems theory, and has published over 180 papers. He is responsible for the development of GIPSY (General Image Processing System), a multiimage processing package which runs on a minicomputer system.

Dr. Haralick is a member of the Association for Computing Machinery, Sigma Xi, the Pattern Recognition Society, and the Society for General Systems Research.

Hyonam Joo was born in Seoul, Korea, on August 2, 1953. He received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 1976 and the M.S. degree in electrical engineering from Virginia Polytechnic Institute and State University, Blacksburg, in 1985. He is currently working toward the Ph.D. degree in electrical engineering from University of Washington.

He worked with the Agency for Defense Development, Korea, as a Research Engineer from

1976–1982. From 1985–1987 he worked with Machine Vision International, Ann Arbor, MI, as a Research Engineer, and from 1987–1989 he worked with Boeing High Technology Center, Seattle, WA, as a Software Engineer. In addition to his graduate work, he is currently a Consultant to both Boeing High Technology Center and NeoPath Inc., Seattle, WA.

Chung Nan Lee received the B.S. and M.S. degrees in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 1980 and 1982, respectively.

From 1984–1986, he was a lecturer in the Department of Electrical Engineering, National Sun Yet-Sen University, Kaoshiung, Taiwan. In 1987–1988 he became System Manager of the Intelligent System Laboratory, while pursuing his graduate studies at the University of Washington, Seattle. Currently he is working toward the Ph.D. degree in electrical engineering. His research interests are in computer vision, image processing, expert systems, robotic navigation, and pattern recognition.

of Michigan, Ann Arbor, granted by Machine Vision International, Ann Arbor, from 1984 to 1985. He was selected as a consultant to the Advisory Group for Aerospace Research and Development, NATO, in 1985. From 1985 to 1986 he was a Visiting Research Professor with the Coordinated Science Laboratory and a Visiting Professor of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign. From 1986 to 1987 he was a Visiting Scientist in the Department of Electrical Engineering, University of Washington, Seattle. His professional interests lie in applied mathematics, image processing, computer and robotic vision, artificial intelligence, and computer architecture. He is a contributor (with E. Ostevold and R. M. Haralick) of the book *Image Recovery: Theory and Application*, edited by Henry Stark; Editor (with R. M. Haralick) of the book *Consistent Labeling Problems in Pattern Recognition*; and author (with T. S. Huang and R. M. Haralick) of the planned book titled *Image Time Sequence Motion Analysis*.

Vinay G. Vaidya, photograph and biography not available at time of publication.

Xinhua Zhuang graduated from Peking University, Peking, China, in 1963, after a four-year undergraduate program and a two-year graduate program in mathematics.

Before 1983 he served as a Senior Research Engineer in the Computing Technique Institute, Hangzhou, China. He was a Visiting Scholar of Electrical Engineering at the Virginia Polytechnic Institute and State University, Blacksburg, from 1983 to 1984, a Visiting Scientist of Electrical and Computer Engineering at the University

Man Bae Kim was born in Seoul, Korea in 1957. He received the B.S. degree in electronics engineering from Hanyang University, Korea in 1983, and the M.S. degree in electrical engineering from the University of Washington, Seattle, in 1986. He is currently a Ph.D. student, and his dissertation topic is on the robust estimation of object pose using robust statistics.