Pose Estimation from Corresponding Point Data

Robert M. Haralick, Hyonam Joo, Chung-nan Lee,
Xinhua Zhuang, Vinay G. Vaidya, and Man Bae Kim
University of Washington

Abstract

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.

Solutions for four different pose estimation problems are presented. Closed form least squares solutions are given to the over constrained 2D–2D and 3D–3D pose estimation problems. A globally convergent iterative technique is given for the 2D perspective projection–3D pose estimation problem. A simplified linear solution and a robust solution to the 2D perspective projection–2D 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 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 signal to noise ratio is less than 40 db. The experiment results also show that robust technique can suppress the blunder data which come from outlier or mismatch.

ISBN 0-12-266719-0

1 Introduction

There are four pose estimation problems with point data. Each arises from two views taken of the same object which 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 3D data relative to the model reference frame. In motion estimation and structure from motion problems there is a rigid body motion of the sensor, the object or both. 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 which uses the corresponding point data set to estimate the translation and rotation which 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 3D 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 3D objects are viewed with a range finder sensor. In the most difficult pose estimation problems, one data set consists of 2D perspective projection of 3D points and the other data set consists of either a 3D point data set, in which case it is known as absolute orientation problem, or the other data set consists of a second 2D perspective projection view of the same 3D 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 multi-camera 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 decreases below 40 db, the mean error skyrockets in the more complex pose estimation problem unless there are hundreds of correspond-

ing 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 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 which 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 which 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

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

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 one, 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 its performance as a baseline reference, put the solution technique in an iterative reweighted form, and finally evaluate the performance using

non–normal noise such as slash noise. This paper represents some initial steps in this strategy.

Section 2 derives a closed form least squares solution to the pure 2D-2D pose estimation problem. And subsequently we derive an iterative weighted least squares solution using a robust method. Section 3 derives a closed form least squares solution to the pure 3D-3D pose estimation problem using a singular value decomposition technique. The least squares solution for both the 2D–2D and 3D–3D pose estimation problems are constrained to produce rotation matrices which are guaranteed to be orthonormal. Section 4 discusses an iterative solution to the 2D perspective projection 3D pose estimation problem. The technique appears to be globally convergent from any initial starting value. Section 5 discusses a solution to the 2D perspective projection–2D perspective projection pose estimation problem. The robust algorithm is also presented.

2 2D–2D Estimation

There are a variety of model based inspection tasks which 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 which 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 two-dimensional rotation and translation.

As mentioned in Section 1, in the matching process, the noise is be 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 correspond to an incorrect point of

the image. (These incorrect points will be called "outliers" through the report.) The outliers may affect the accuracy and stability of the pose estimation.

We have recognized that some data points, which arise from heavily tailed distribution 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 2D–2D pose estimation is organized as follows. Section 2.1 gives a precise statement of this problem as a weighted least squares problem. In Section 2.2, 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 2.4, 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 2D–2D pose estimation.

2.1 Statement of Problem

In the simple two-dimensional pose detection problem, we are given N two-dimensional coordinate observations from the observed image: $x_1, ..., x_N$. 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: $y_1, ..., y_N$. 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 po-

sition 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 two-dimensional 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 $y_n = Rx_n + t$, $n = 1, ..., N$. Since there are likely to be small observational errors, the real problem must be posed as a minimization. Determine R and t which 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 $w_n, n = 1, ..., N$ satisfy $w_n \geq 0$ 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: $w_n = 1/N$.

2.2 Least Squares Method

Upon expanding Eq. 1 out we have

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

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

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

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

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

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.

$$\begin{aligned}
\epsilon^2 &= \sum_{n=1}^{N} w_n \Big[(y_n - (\bar{y} - R\bar{x}))'(y_n - (\bar{y} - R\bar{x})) \\
&\quad -2(y_n - (\bar{y} - R\bar{x}))' R x_n + x_n' x_n \Big] \\
&= \sum_{n=1}^{N} w_n \Big[(y_n - \bar{y})'(y_n - \bar{y}) + 2(y_n - \bar{y})' R\bar{x} + \bar{x}' R' R \bar{x} \\
&\quad -2(y_n - \bar{y})' R x_n - 2\bar{x} R' R x_n + x_n' x_n \Big] \\
&= \sum_{n=1}^{N} w_n \Big[(y_n - \bar{y})'(y_n - \bar{y}) \\
&\quad -2(y_n - \bar{y})' R(x_n - \bar{x}) + (x_n - \bar{x})'(x_n - \bar{x}) \Big]
\end{aligned} \tag{7}$$

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

$$x_n = \begin{pmatrix} x_{1n} \\ x_{2n} \end{pmatrix}, \quad y_n = \begin{pmatrix} y_{1n} \\ y_{2n} \end{pmatrix}$$

$$\bar{x} = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix}, \quad \text{and} \quad \bar{y} = \begin{pmatrix} \bar{y}_1 \\ \bar{y}_2 \end{pmatrix}$$

then

$$\begin{aligned}(y_n - \bar{y})' R (x_n - \bar{x}) = \\ (y_{n1} - \bar{y}_1)\cos\theta(x_{n1} - \bar{x}_1) + (y_{n1} - \bar{y}_1)(-\sin\theta)(x_{n2} - \bar{x}_2) \\ +(y_{n2} - \bar{y}_2)\sin\theta(x_{n1} - \bar{x}_1) + (y_{n2} - \bar{y}_2)\cos\theta(x_{n2} - \bar{x}_2) \quad (9)\end{aligned}$$

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

$$\begin{aligned} 0 \;=\; & -2\sum_{n=1}^{N} w_n \Big[(y_{n1} - \bar{y}_1)(-\sin\theta)(x_{n1} - \bar{x}_1) \\ & +(y_{n1} - \bar{y}_1)(-\cos\theta)(x_{n2} - \bar{x}_2) \\ & +(y_{n2} - \bar{y}_2)\cos\theta(x_{n1} - \bar{x}_1) + \\ & (y_{n2} - \bar{y}_2)(-\sin\theta)(x_{n2} - \bar{x}_2)\Big] \end{aligned} \quad (10)$$

Letting

$$A = \sum_{n=1}^{N} w_n \left[(y_{n1} - \bar{y}_1)(x_{n1} - \bar{x}_1) + (y_{n2} - \bar{y}_2)(x_{n2} - \bar{x}_2)\right] \quad (11)$$

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

Then

$$0 = A\sin\theta + B\cos\theta$$

Hence,

$$\cos\theta = \frac{-A}{\sqrt{A^2 + B^2}} \quad \text{and} \quad \sin\theta = \frac{B}{\sqrt{A^2 + B^2}} \quad (12)$$

or

$$\cos\theta = \frac{A}{\sqrt{A^2 + B^2}} \quad \text{and} \quad \sin\theta = \frac{-B}{\sqrt{A^2 + B^2}} \quad (13)$$

The correct value for θ will, in general, be unique and will be that θ which 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.

2.3 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.

M estimator

In M estimator, the solution of an 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 a sample size. ρ is an arbitrary non-negative monotonically increasing function (called the object function). $\psi(x_i - \theta)$ is a derivative of $\rho(x_i - \theta)$ with respect to θ and is called an M-estimator.

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

Eq. 15 can be written equivalently as

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

where

$$w_i = \frac{\psi(x_i - \theta)}{x_i - \theta}, \quad i = 1, \ldots, N \tag{18}$$

This gives a formal representation of θ as a weighted mean

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

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

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

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

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

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

The weight function of Tukey's biweight is

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

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

Iterative weighted least squares method

The residual error ϵ_i for nth data sample is

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

where $i = 1, .., N$. 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 $i = 1, ..., N$.

- Step.1 Select initial starting values for R and t.
- Step.2 R and t give weights w_i where $i = 1, ..., N$. To find weights, we use Eq. 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=$1, .., N$. Thus, w_i is expressed as

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

  The new R and t are obtained from the new weights.

- Step.3 If some degree of convergence in R and t are obtained, go to Step.4. If not go to Step. 2
- Step.4 From the final W, we normalize the weights and find estimates for rotation angle and translation using the solution derived in Section 2.2. Stop.

2.4 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 signal to noise ratio (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 which is Gaussian the normalized interquartile range is just the noise standard deviation. For distributions such as the slash distribution which 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 Figures 1, 2, and 3. Figure 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 1 degree while for 100 corresponding point pairs the SNR can go as low as 25 db while maintaining a less than 1 degree mean absolute rotation error. The pattern for mean translational distance error is similar. This is shown in Figure 2. To maintain a mean translational distance error of .01, which is a relative error of about .25%, requires 100 corresponding point pairs at a 32 db SNR. Using only 8 corresponding point pairs, even a SNR of 52 db provides a mean translation distance error of about .03, or .75%. Figure 3 shows rotational error as a function of number of corresponding point pairs for a few values of SNR. Figure 3 suggests a rapid increase in expected error when there are fewer than 50 corresponding point pairs for a 35 db SNR. In the next experiments, we examine the performance of the least squares and robust methods with outliers present in the

image. To generate the outliers, we intentionally moved 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 outliers was varied from 10 % to 50%. Figs. 4 and 5 show the mean rotational and translational errors as a function of the SNR for the PO (percentage of the outliers) varying between 10 % and 50 % 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 non-outliers. Figs. 6 and 7 show the mean rotational and translational errors. The number of corresponding point pairs is changed from 20 of Figs. 4 and 5 to 50. Other parameters are unchanged. The robust method shows better performance than the least squares method. We can notice the improved performance in the translational error. Figs. 8 and 9 show the mean rotational and translational errors as a function of the number of corresponding point pairs for the PO varying between 10 % and 50 % when the least squares and robust methods are used. The SNR is fixed 10 db. As we increase the PO, the performance is degraded. The least squares method experiences unstability when the PO is 40 %. This unstability is common in the least squares method. But, we do not observe any severe unstability in the robust method, which provides us with reliable results. Since the SNR is 10 *db*, the performances of the robust method does not show large improvement compared with the least squares method. In Figs. 10 and 11, we changed the SNR from 10 db of Figs. 8 and 9 to 30 db and observed the performance. The other parameters are unchanged. Since we has increased the SNR, the robust method clearly shows better performance and stability than the least squares method. In Figs. 12 and 13, we compare the mean rotational and translational errors of the least squares and robust methods. The

number of corresponding point pairs is varied between 10 and 50. The PO is fixed 20 %. We changed the SNR from 10 db to 50 db. The rotational and translational errors of the robust method show better performance than those of the least squares method. When the SNR is 10 *db*, the performance of the two methods does not show much difference as observed before. Finally, we tried an interesting experiment. All outliers were intentionally removed from the data sets. Then, we applied the equal weight least squares method. Fig. 14 shows the performance of the mean rotational and translational errors as a function of the SNR for different POs. The number of the corresponding point pairs is 20. Comparing it with Figs. 4 and 5, we observe that the performance of this experment is better than that of the robust method. This observation reveals some interesting properties of the robust method. First, in the robust method, the outliers are not perfectly eliminated. Ideally, the weights assigned to the outliers should be zero or very small. But, in practice, some outliers have significant weights, which indicates that the robust method does not detect such outliers. Second, there are some data points which are not outliers and have weights which indicates that they are considered as outliers. For example, during the experiment, some data points having large noise were detected as outliers. Finally, in computing the rejection point, cS of the Tukey's biweight, we adopted 6.0 for the tuning constant, c. But, the optimum c should be found. However, since it involves the complicated mathematics, we did not take it into account in the analysis.

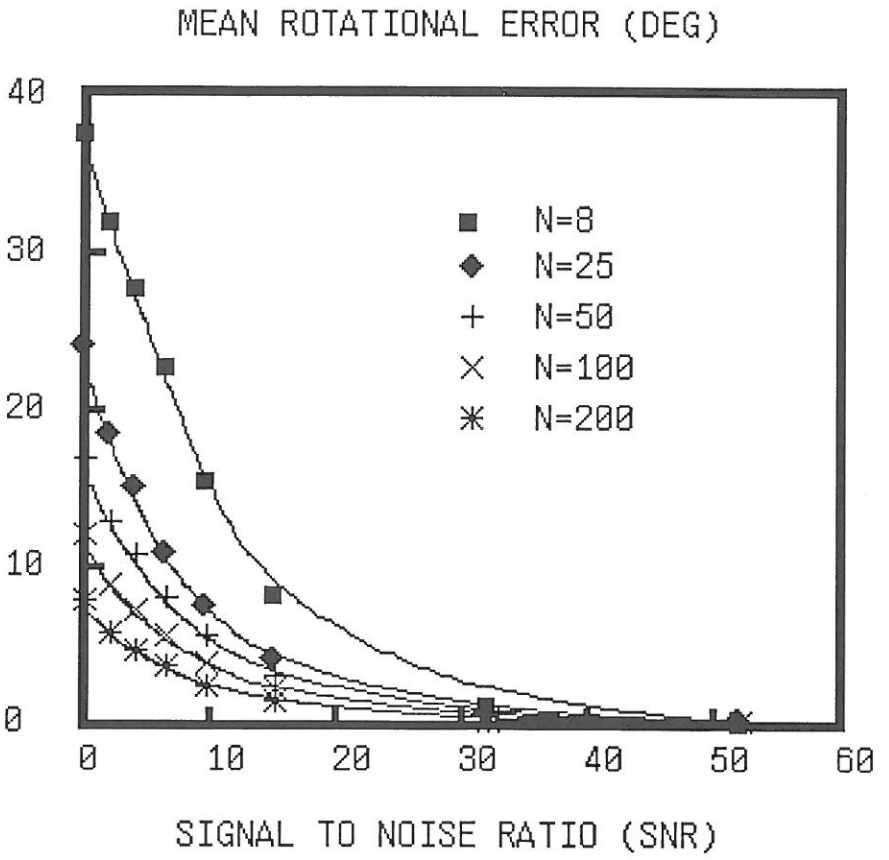

Figure 1: Mean absolute rotational error as a function of signal to noise ratio for the 2D–2D pose estimation problem.

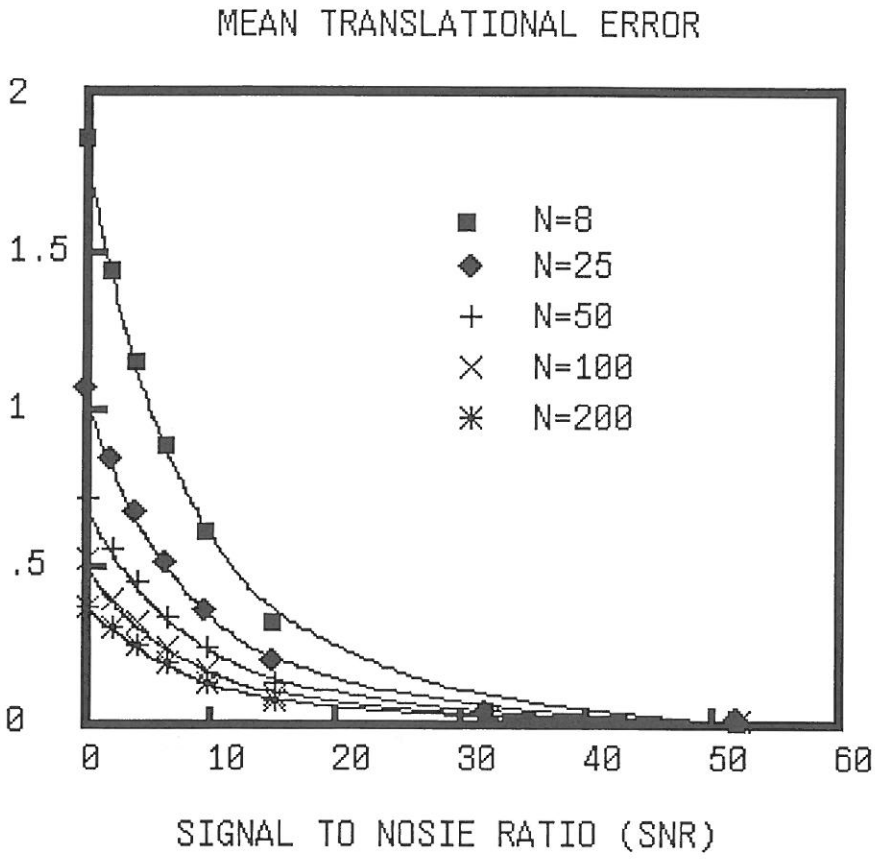

Figure 2: Mean translational distance error as a function of signal to noise ratio for the 2D–2D pose estimation problem.

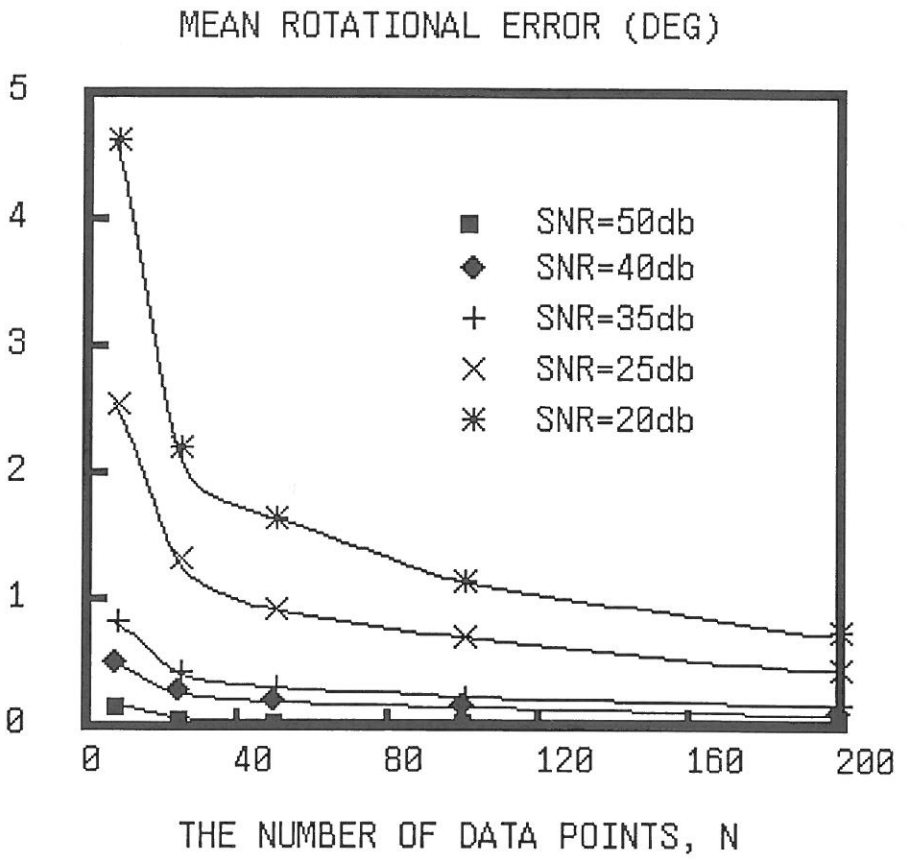

Figure 3: Mean absolute rotational error as a function of the number of corresponding point pairs for the 2D–2D pose estimation problem.

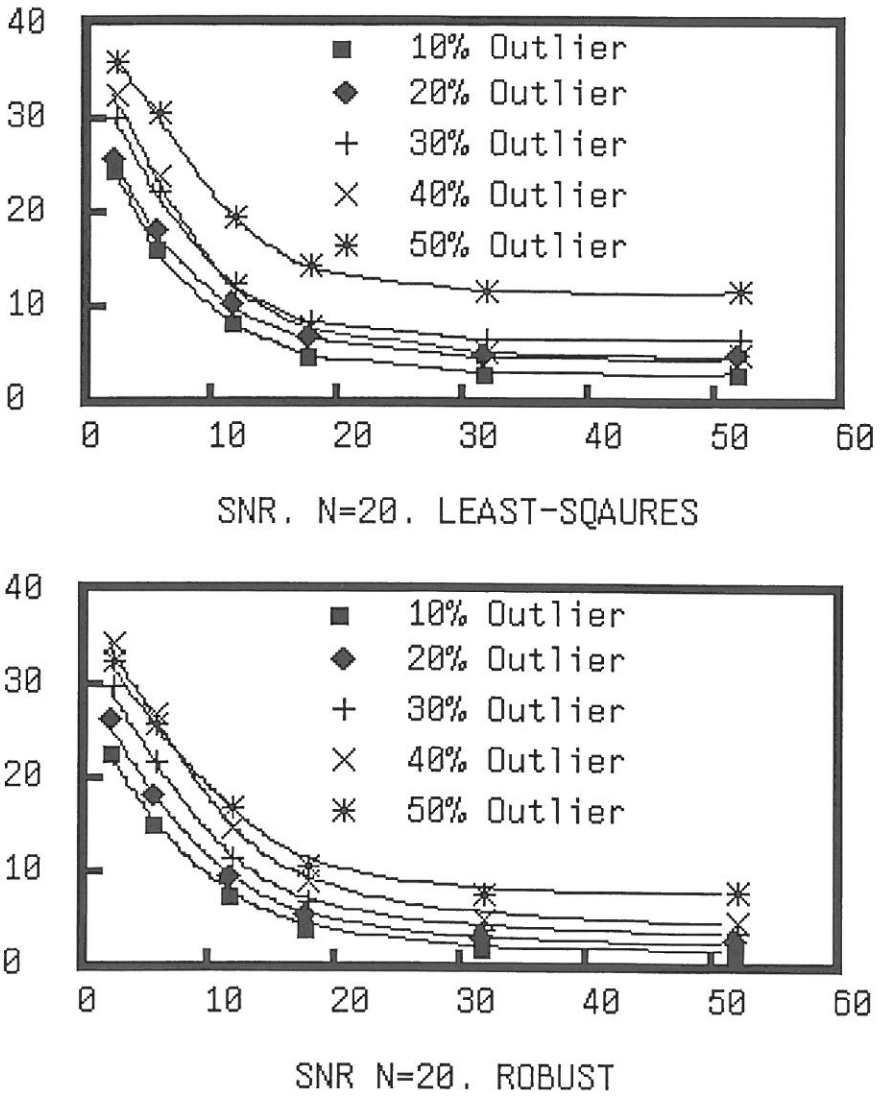

Figure 4: Mean absolute rotational error of least squares method and robust method as a function of SNR for the 2D-2D pose estimation problem. The number of corresponding point pairs is 20. The percentage of the outliers is changed from 10% to 50%.

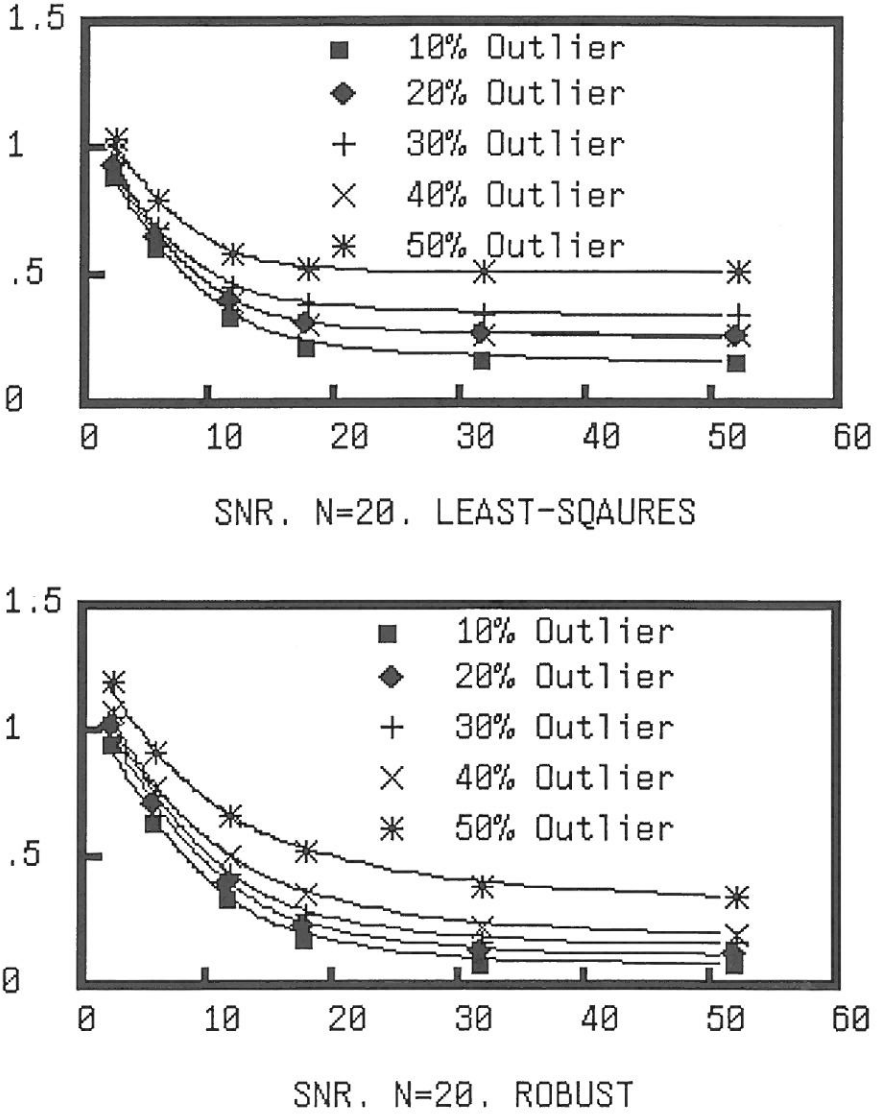

Figure 5: Mean translational distance error of least squares method and robust method as a function of SNR for the 2D-2D pose estimation problem. The number of corresponding point pairs is 20. The percentage of the outliers is changed from 10% to 50%.

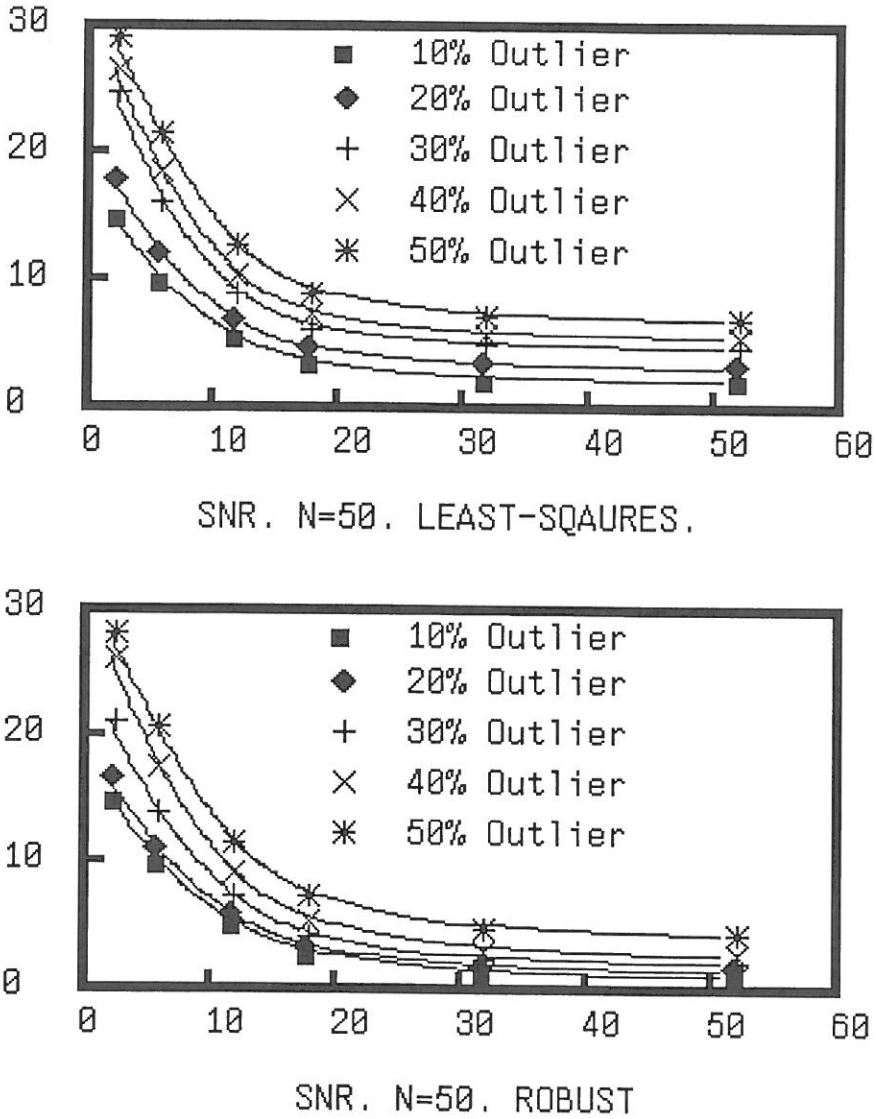

Figure 6: Mean absolute rotational error of least squares method and robust method as a function of SNR for the 2D-2D pose estimation problem. The number of corresponding point pairs is 50. The percentage of the outliers is changed from 10% to 50%.

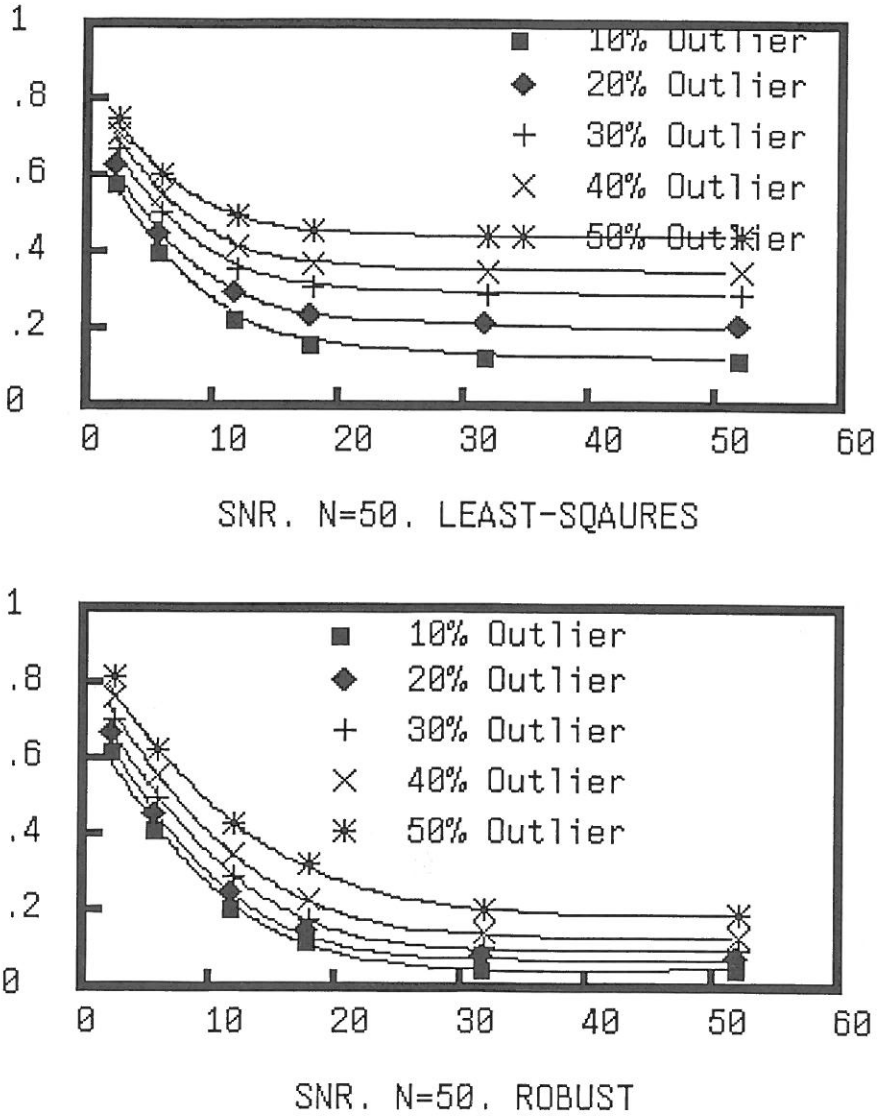

Figure 7: Mean translational distance error of least squares method and robust method as a function of SNR for the 2D-2D pose estimation problem. The number of corresponding point pairs is 50. The percentage of the outliers is changed from 10% to 50%.

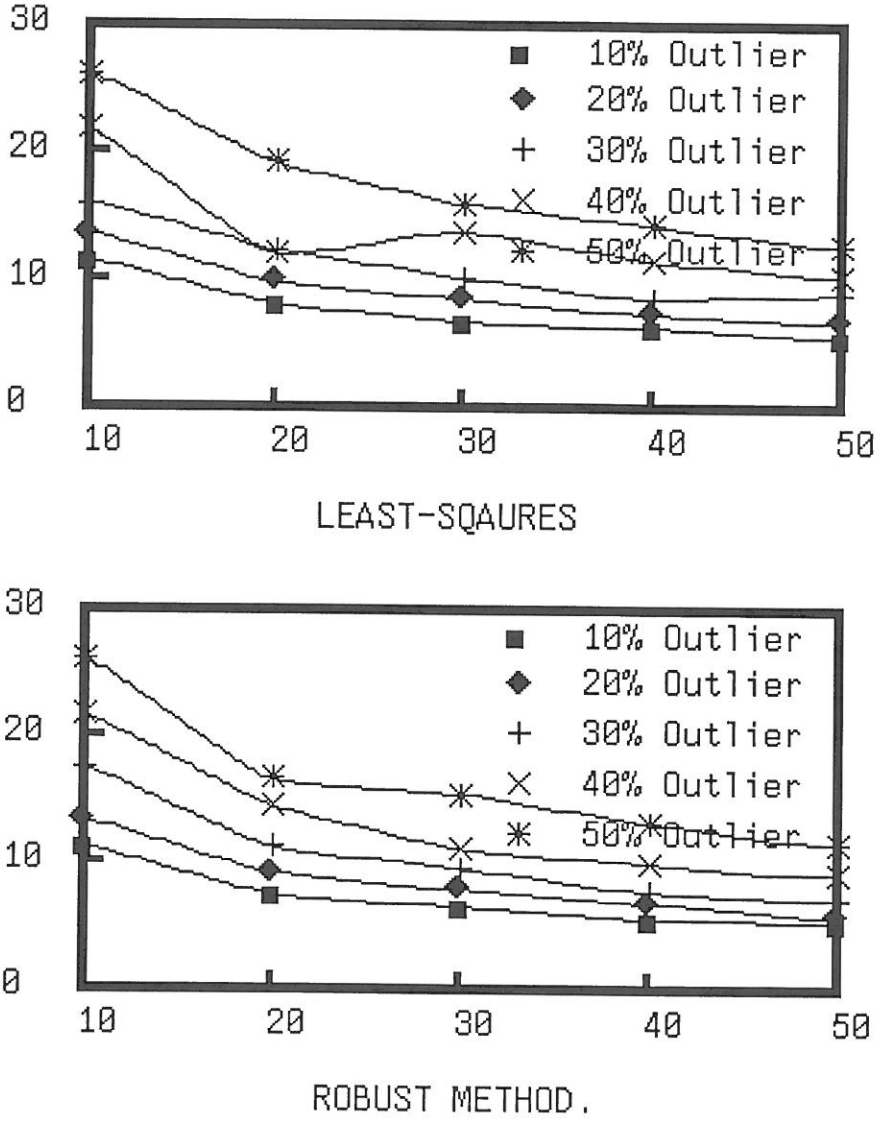

Figure 8: Mean absolute rotational error of least squares method and robust method as a function of corresponding point pairs for the 2D-2D pose estimation problem. The SNR is 10db. The percentage of the outliers is changed from 10% to 50%.

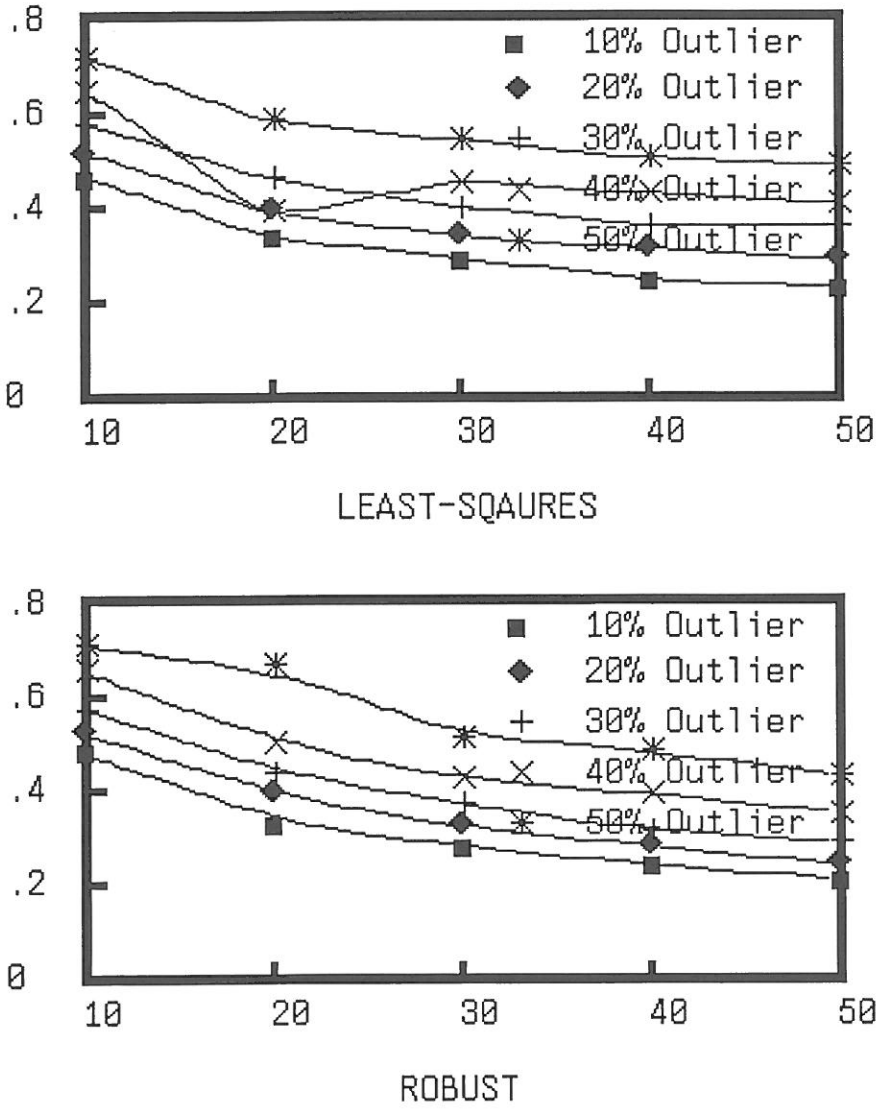

Figure 9: Mean translational distance error of least squares method and robust method as a function of corresponding point pairs for the 2D-2D pose estimation problem. The SNR is 10db. The percentage of the outliers is changed from 10% to 50%.

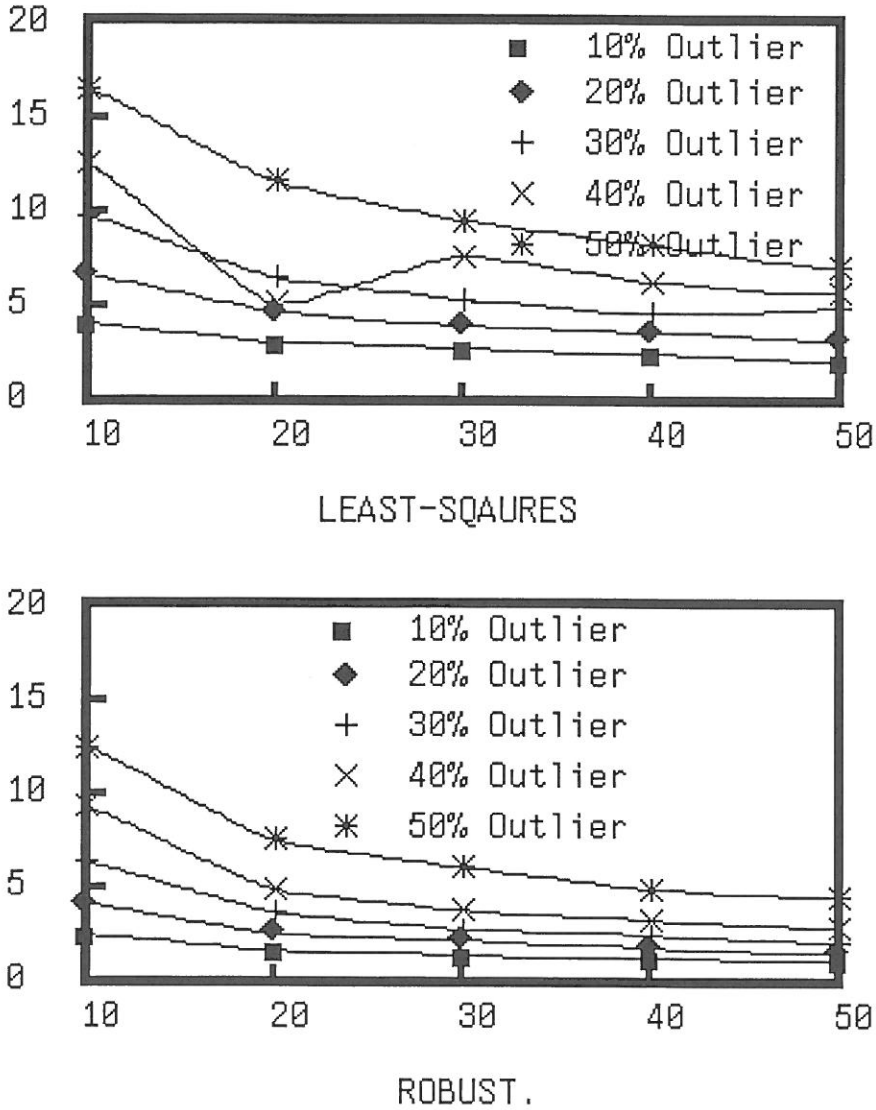

Figure 10: Mean absolute rotational error of least squares method and robust method as a function of corresponding point pairs for the 2D-2D pose estimation problem. The SNR is 30db. The percentage of the outliers is changed from 10% to 50%.

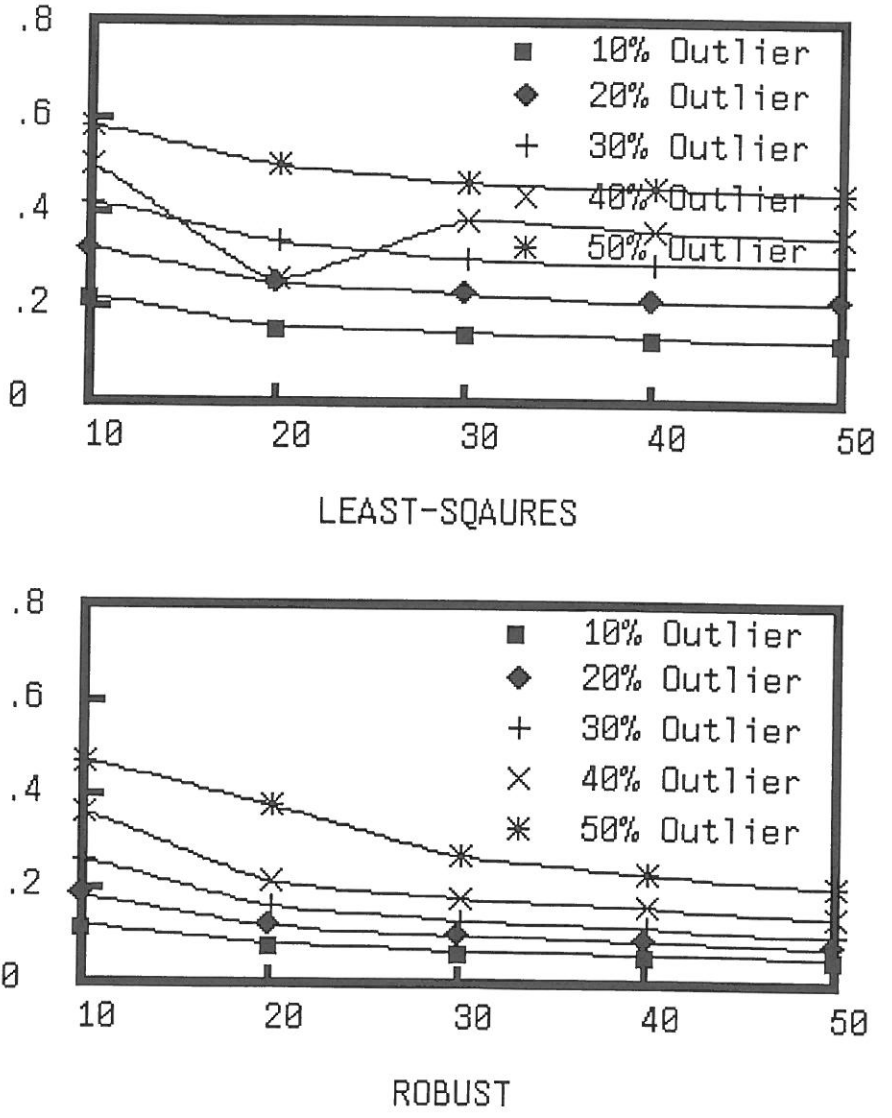

Figure 11: Mean translational distance error of least squares method and robust method as a function of corresponding point pairs for the 2D-2D pose estimation problem. The SNR is 30db. The percentage of the outliers is changed from 10% to 50%.

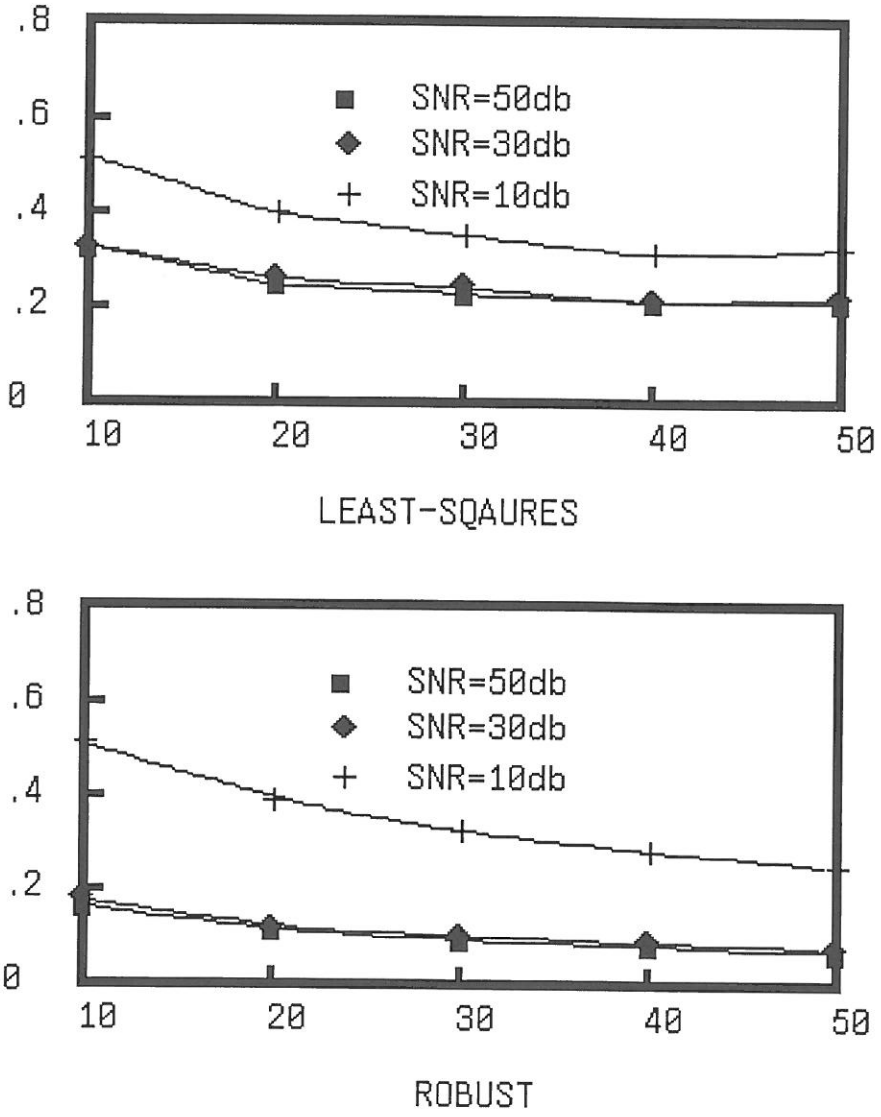

Figure 12: Mean absolute rotational error of the least squares and robust methods as a function of the number of corresponding point pairs for 2D-2D pose estimation problem. SNR is changed from 10 db to 50 db. The percentage of the outliers is fixed 20 %.

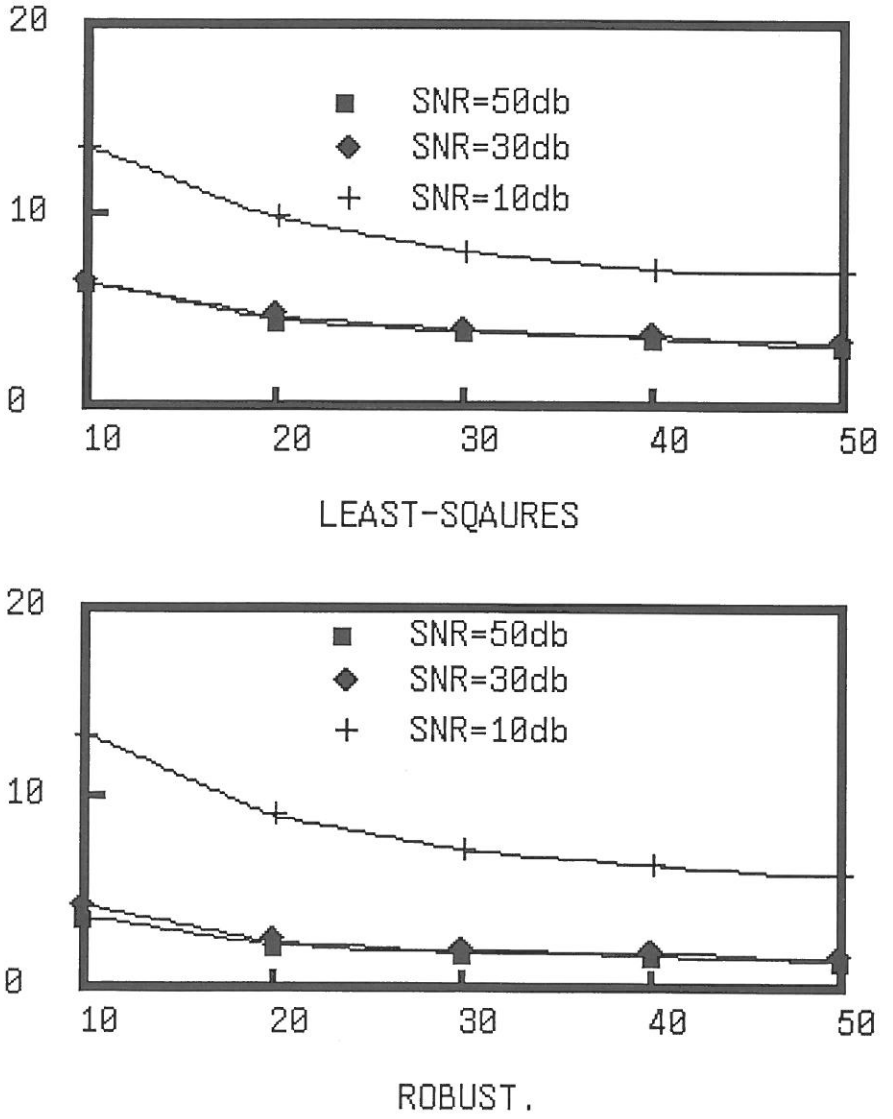

Figure 13: Mean translational distance error of the least squares and robust methods as a function of the number of corresponding point pairs for 2D-2D pose estimation problem. SNR is changed from 10 db to 50 db. The percentage of the outliers is fixed 20 %.

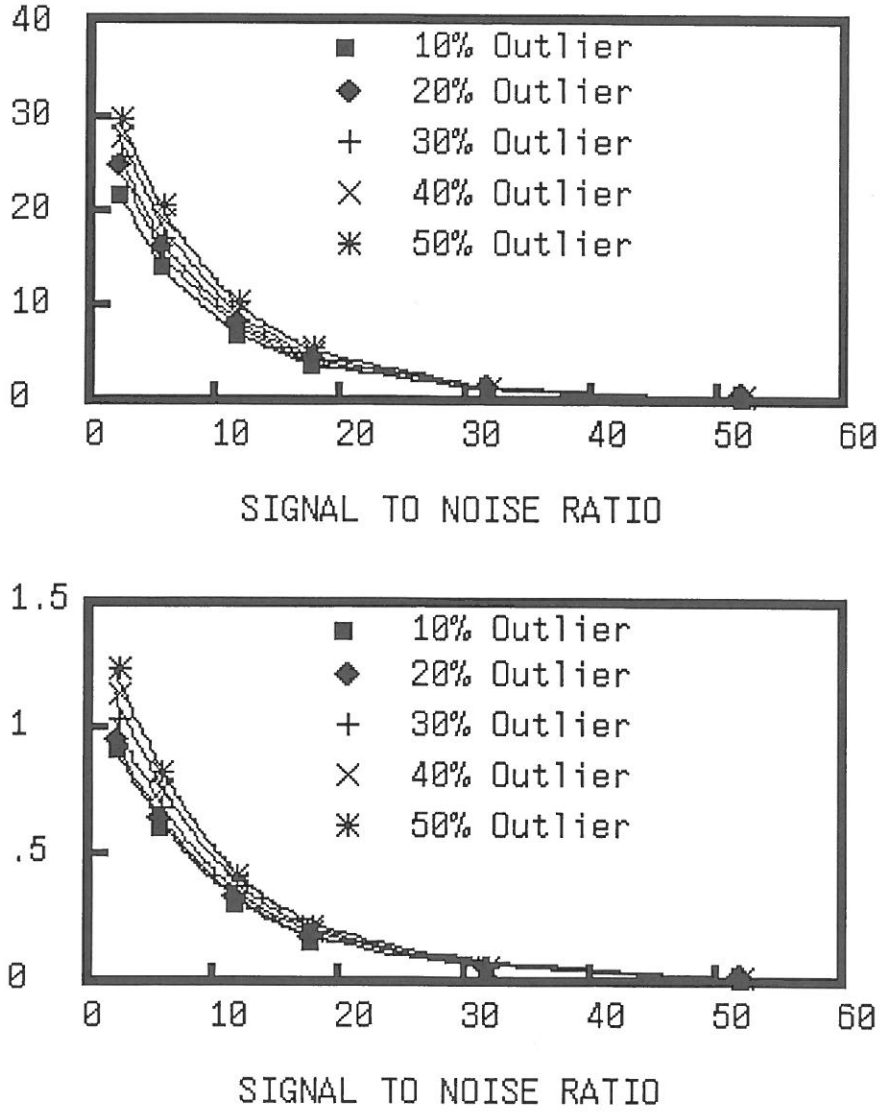

Figure 14: Mean absolute rotational and translational distance errors of the least squares method as a function of SNR for the 2D-2D pose estimation problem. The outliers are removed and the least squares method is applied. The number of corresponding point pairs is 20. The percentage of the outliers is changed from 10% to 50% with a scale of 20 %.

3 3D–3D Estimation

3.1 Statement of Problem

Let $y_1 ... y_N$ be N points in Euclidean 3-space. Let R be a rotation matrix and t be a translation vector. Let $x_1, ..., x_N$ be the points in Euclidean 3-space which match $y_1, ..., y_N$. 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 3D–3D pose estimation problem is to infer R and t from $x_1, ..., x_N$ and $y_1, ..., y_N$.

3.2 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

$$R = \begin{pmatrix} r_1' \\ r_2' \\ r_3' \end{pmatrix} \text{ where each } r_i \text{ is a } 3 \times 1 \text{ vector}$$

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

$$\begin{aligned} r_1' r_1 &= 1 \\ r_2' r_2 &= 1 \\ r_3' r_3 &= 1 \\ r_1' r_2 &= 0 \\ r_1' r_3 &= 0 \\ r_2' r_3 &= 0 \end{aligned} \tag{25}$$

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

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

$$+\sum_{k=1}^{3}\lambda_k(r_k'r_k-1)+2\lambda_4 r_1'r_2$$
$$+2\lambda_5 r_1'r_3+2\lambda_6 r_2'r_3 \tag{26}$$

$$x_n=\begin{pmatrix}x_{n1}\\x_{n2}\\x_{n3}\end{pmatrix},\quad y_n=\begin{pmatrix}y_{n1}\\y_{n2}\\y_{n3}\end{pmatrix},\text{ and }\quad t=\begin{pmatrix}t_1\\t_2\\t_3\end{pmatrix}$$

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

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

Setting these partials to zero results in

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

By rearranging we obtain

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

$$\text{where }\overline{x}=\frac{\sum_{n=1}^{N}w_nx_n}{\sum_{n=1}^{N}w_n}\text{ and }\overline{y}=\frac{\sum_{n=1}^{N}w_ny_n}{\sum_{n=1}^{N}w_n}.$$

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

$$\epsilon^2=\sum_{n=1}^{N}w_n\sum_{k=1}^{3}\left(y_{nk}-\overline{y}_n-r_k'(x_n-\overline{x})\right)^2+\sum_{k=1}^{3}\lambda_k(r_k'r_k-1)$$
$$+2\lambda_4 r_1'r_2+\lambda_5 r_1'r_3+\lambda_6 r_2'r_3 \tag{28}$$

where

$$\overline{x}=\begin{pmatrix}\overline{x}_1\\\overline{x}_2\\\overline{x}_3\end{pmatrix},\quad\overline{y}=\begin{pmatrix}\overline{y}_1\\\overline{y}_2\\\overline{y}_3\end{pmatrix}.$$

Now we take partial derivatives of ϵ^2 with respect to the components of each y_n. To write things more compactly, by $\frac{\partial\epsilon^2}{\partial r_n}$ 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,

$$\frac{\partial \epsilon^2}{\partial r_1} = \sum_{n=1}^{N} 2w_n\Big(y_{n1} - \overline{y}_1 - r_1'(x_n - \overline{x})\Big)\,(x_n - \overline{x})(-1) + 2\lambda_1 r_1 + 2\lambda_4 r_2 + 2\lambda_5 r_3 \tag{29}$$

$$\frac{\partial \epsilon^2}{\partial r_2} = \sum_{n=1}^{N} 2w_n\Big(y_{n2} - \overline{y}_2 - r_2'(x_n - \overline{x})\Big)\,(x_n - \overline{x})\,(-1) + 2\lambda_2 r_2 + 2\lambda_4 r_1 + 2\lambda_6 r_3 \tag{30}$$

$$\frac{\partial \epsilon^2}{\partial r_3} = \sum_{n=1}^{N} 2w_n\Big(y_{n3} - \overline{y}_3 - r_3'(x_n - \overline{x})\Big)\,(x_n - \overline{x})\,(-1) + 2\lambda_3 r_3 + 2\lambda_5 r_1 + 2\lambda_6 r_2 \tag{31}$$

Setting these partial derivatives to zero and rearranging we obtain

$$\sum_{n=1}^{N} w_n(x_n - \overline{x}(x_n - \overline{x})' r_1 + \lambda_1 r_1 + \lambda_4 r_2 + \lambda_5 r_3 = \sum_{n=1}^{N} w_n(y_{n1} - \overline{y}_1)(x_n - \overline{x}) \tag{32}$$

$$\sum_{n=1}^{N} w_n(x_n - \overline{x})(x_n - \overline{x})' r_2 + \lambda_4 r_1 + \lambda_2 r_2 + \lambda_6 r_3 = \sum_{n=1}^{N} w_n(y_{n2} - \overline{y}_2)(x_n - \overline{x}) \tag{33}$$

$$\sum_{n=1}^{N} w_n(x_n - \overline{x})(x_n - \overline{x})' r_3 + \lambda_5 r_1 + \lambda_6 r_2 + \lambda_3 r_3 = \sum_{n=1}^{N} w_n(y_{n3} - \overline{y}_3)(x_n - \overline{x}) \tag{34}$$

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

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

$$\text{and } B = (b_1 b_2 b_3) \text{ when } b_k = \sum_{n=1}^{N} w_n (y_{nk} - \overline{y}_k)(x_n - \overline{x})$$

Then Eq. 33 can be simply rewritten as

$$A\,R' + R'\Lambda = B \tag{35}$$

Multiplying both sides of Eq. 35 on the left by R we have

$$R\,A\,R' + \Lambda = RB \tag{36}$$

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,

$$R\,B = (RB)' \tag{37}$$

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

$$\begin{aligned} RUDV &= (UDV)'R' \\ &= V'DU'R' \end{aligned} \tag{38}$$

By observation, a solution for R is immediately obtained as

$$R = V'U' \tag{39}$$

Solutions to this problem can be found in the photogrammetry literature beginning with Thompson (1958), Schut (1960), Tienstra (1969), and Pope (1970). Blais (1972) gives a solution to the problem in the case where there may be a scale factor or magnification different than 1. Sansò (1973) gives a solution to the problem using quaternions. Huang (1987) and Haralick et al. (1987) have discussed the singular value decomposition approach to the problem.

3.3 Experimental Results

Over 144,000 simulation experiments were done in which 3D 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 Eq. 27. The rotation and translation was then estimated using Eq. 27 and Eq. 39.

The number of corresponding point pairs was varied between 10 and 200 in 9 steps. The signal to noise ratio, which is defined as 20 log (dynamic range of 3D points/normalized interquartile range of noise), was varied between 28 db and 160 db in 8 steps. The noise distribution type was varied between Gaussian and uniform. For each calculation one thousand trials were run.

Figure 15 illustrates a typical experimental result. It shows the mean angle error of the rotation, in degrees, as a function of signal to noise ratio with Gaussian noise. The plot indicates that when the number of 3D points is 50, then the RMS error of the rotation angle will be less than 3 degrees when the signal to noise ratio is greater than 55 db.

Figure 16 shows the angle error of the translation vector as a function of signal to noise ratio. The plot is comparable to the behavior of that in Figure 4. Figures 6 and 7 show similar experiments using uniform noise.

Figure 19 shows rotation angle error plotted as a function of number of points in the coresponding 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 corrsponding point data pairs is above 40, the estimates improve for increasing- sized sets.

Figure 20 illustrates the translation error angle as a function of the number of corresponding point data pairs for varying signal to noise ratios and Gaussian noise. Figures 10 and 11 show comparable plots for uniform noise.

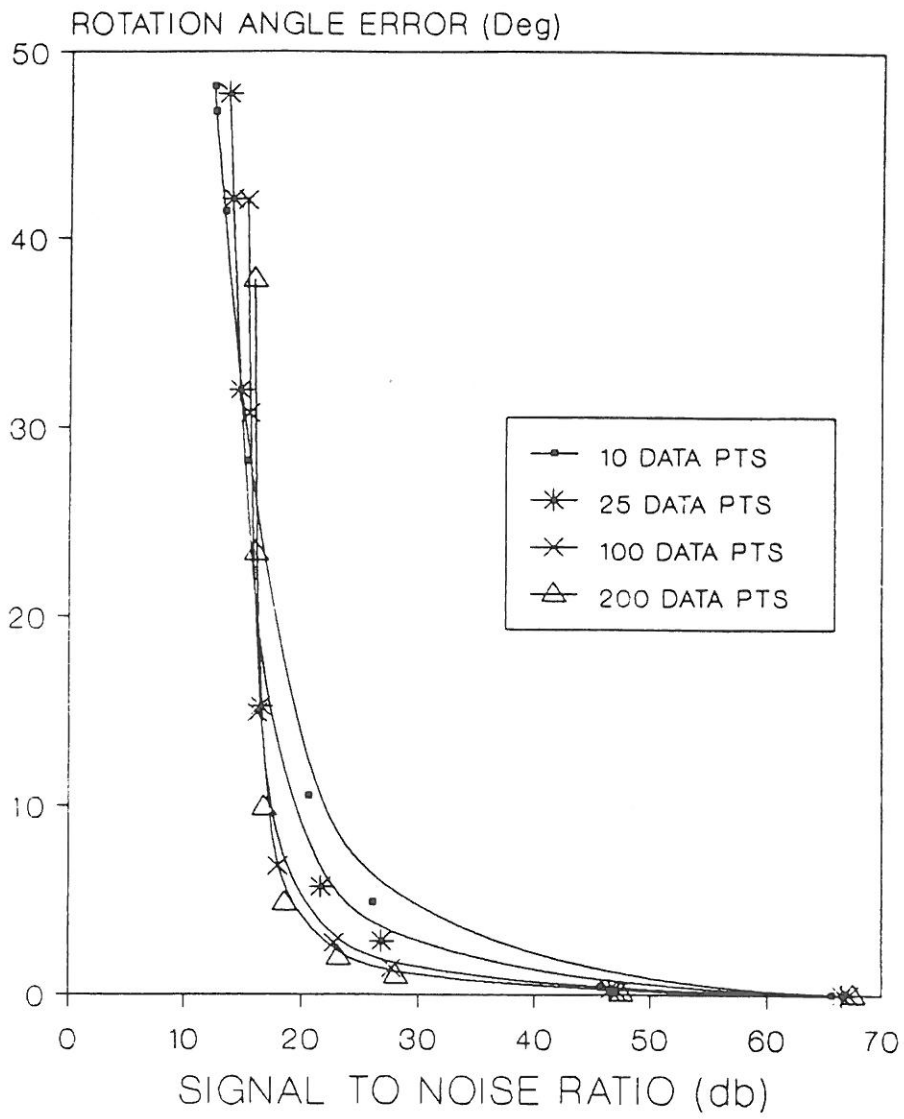

Figure 15: Mean rotation angle error versus signal to noise ratio with Gaussian noise. Corresponding point data set sizes vary between 10 and 200 pairs. Each point on the graph represents 1,000 trials.

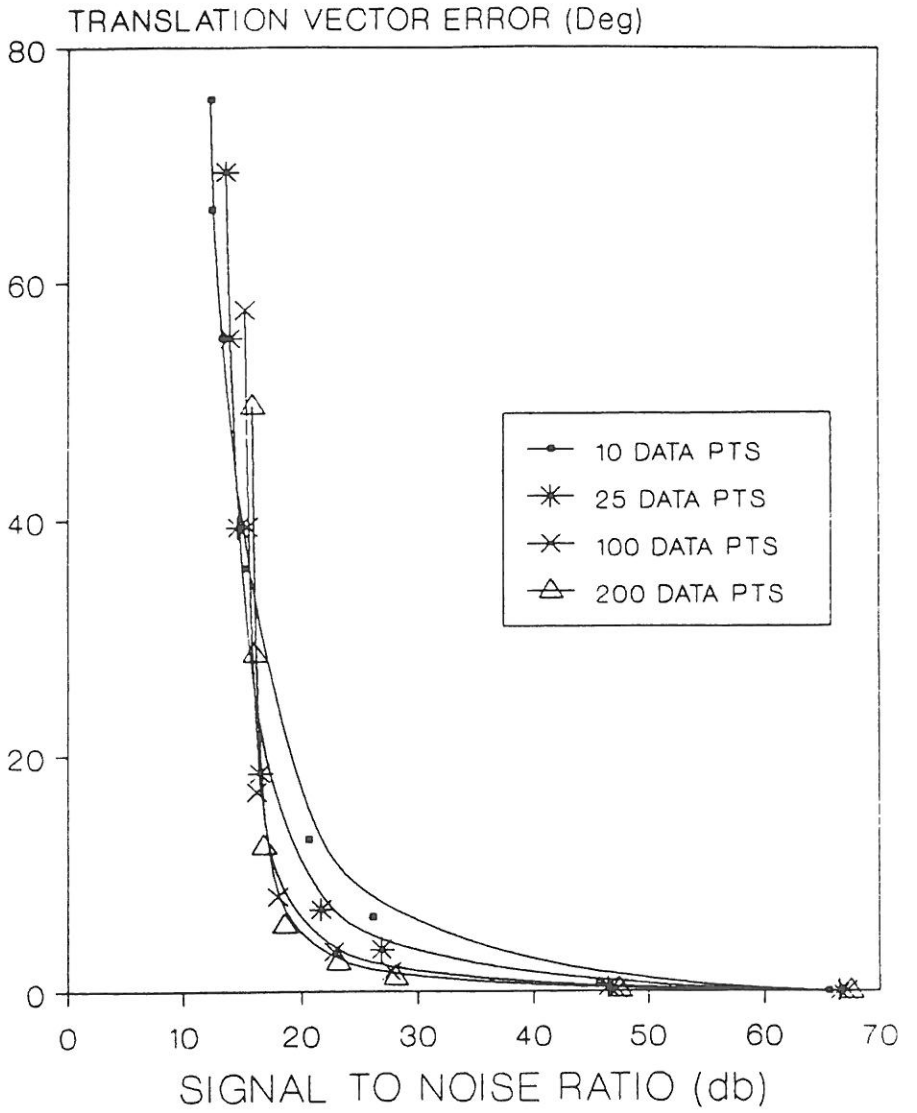

Figure 16: Mean translation vector error versus signal to noise ratio with Gaussian noise. Corresponding point data set sizes vary between 10 and 200 pairs. Each point on the graph represents 1,000 trials.

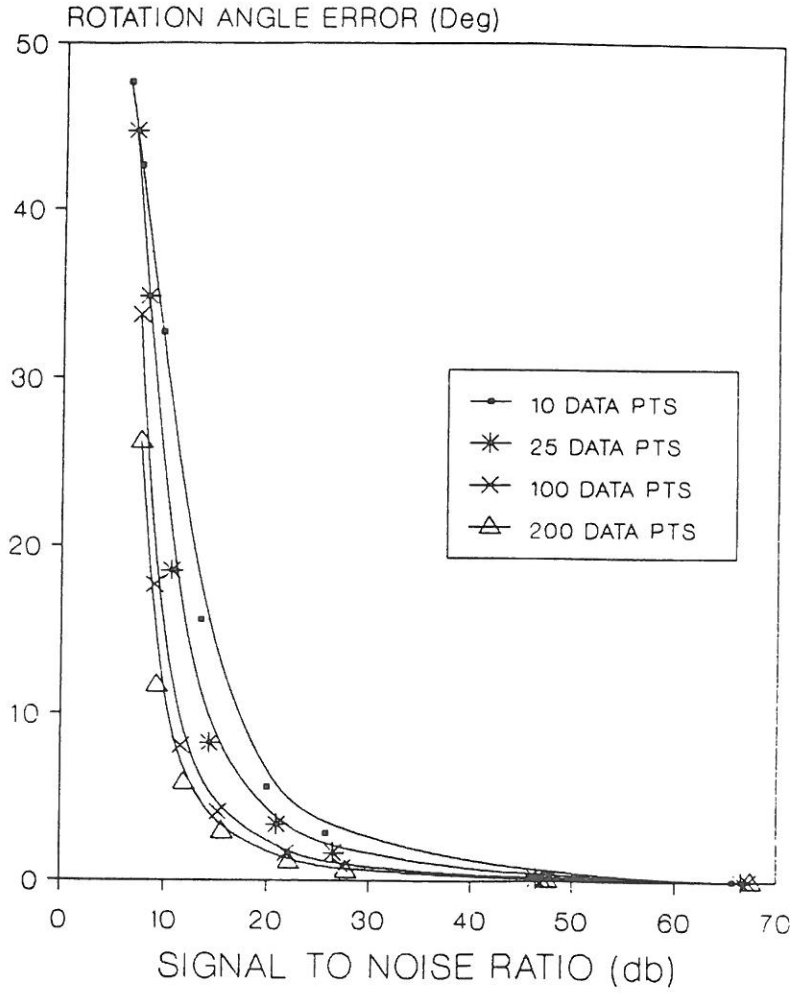

Figure 17: Mean rotation angle error versus signal to noise ratio with uniform noise. Corresponding point data set sizes vary between 10 and 200 pairs. Each point on the graph represents 1,000 trials.

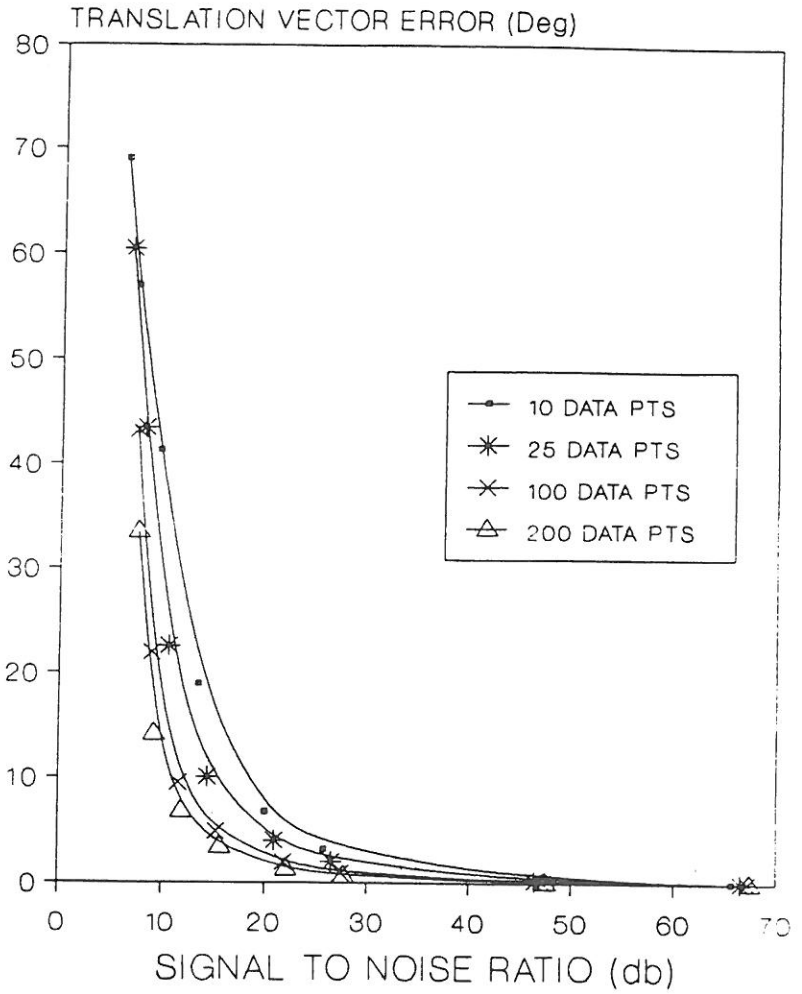

Figure 18: Mean translation vector error versus signal to noise ratio with uniform noise. Corresponding point data set sizes vary between 10 and 200 paris. Each point on the graph represents 1,000 trials.

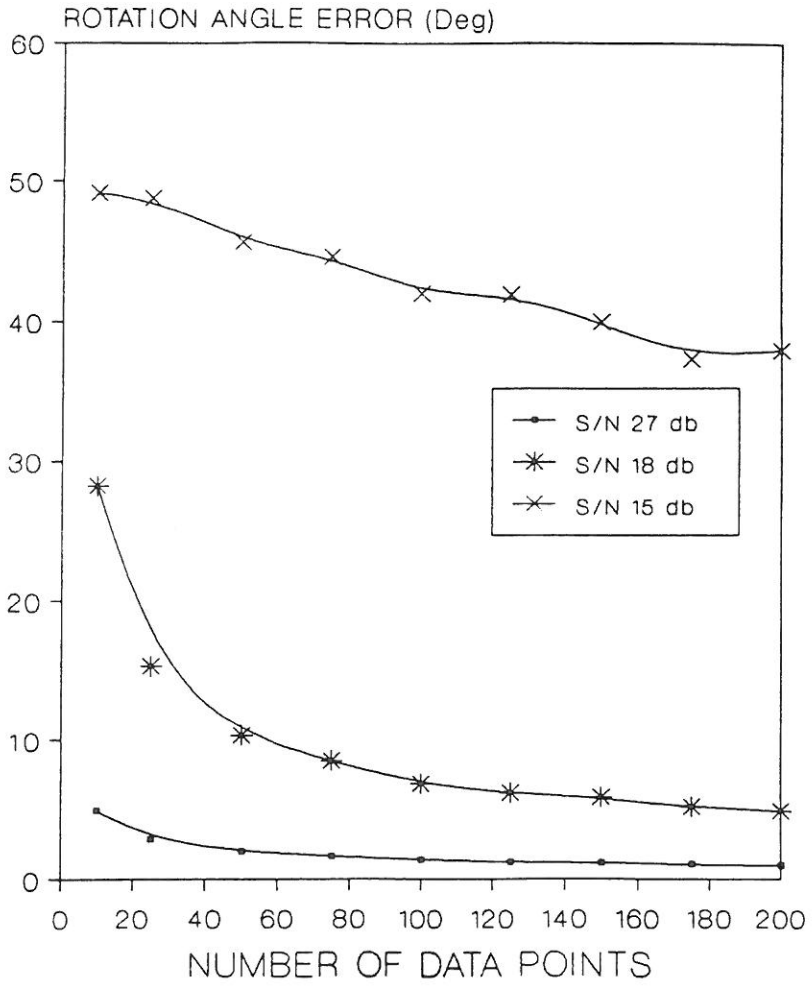

Figure 19: Mean rotation angle error versus number of points with Gaussian noise.

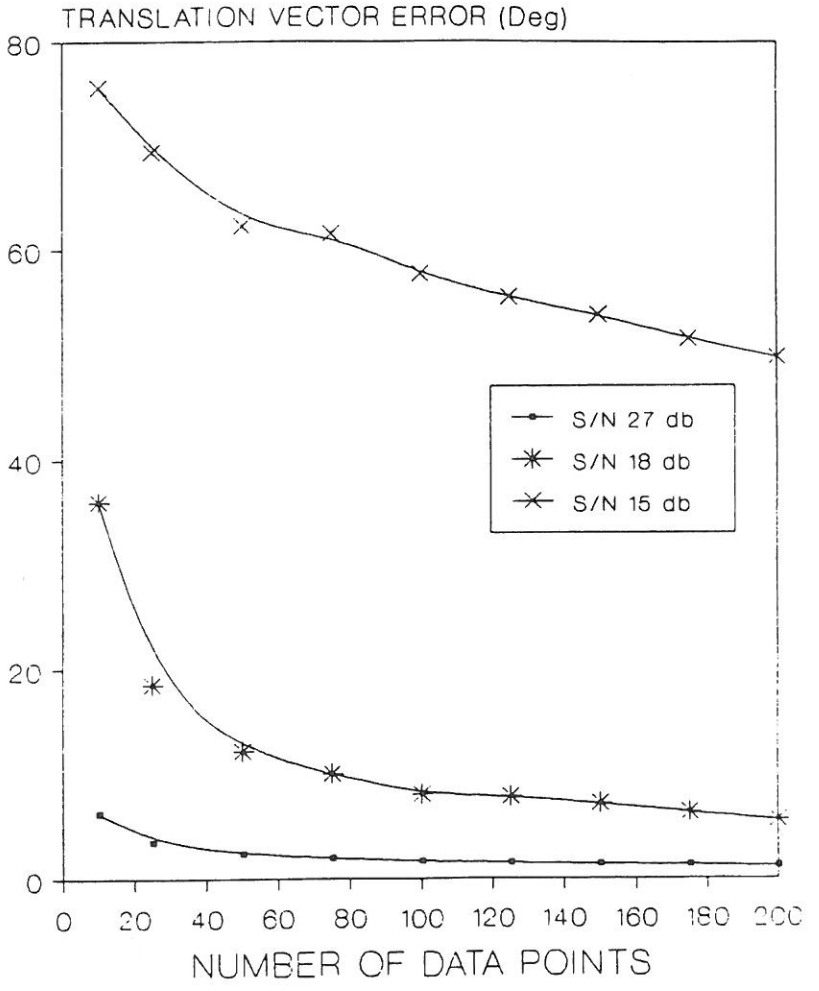

Figure 20: Mean translation angle error versus number of points with Gaussian noise.

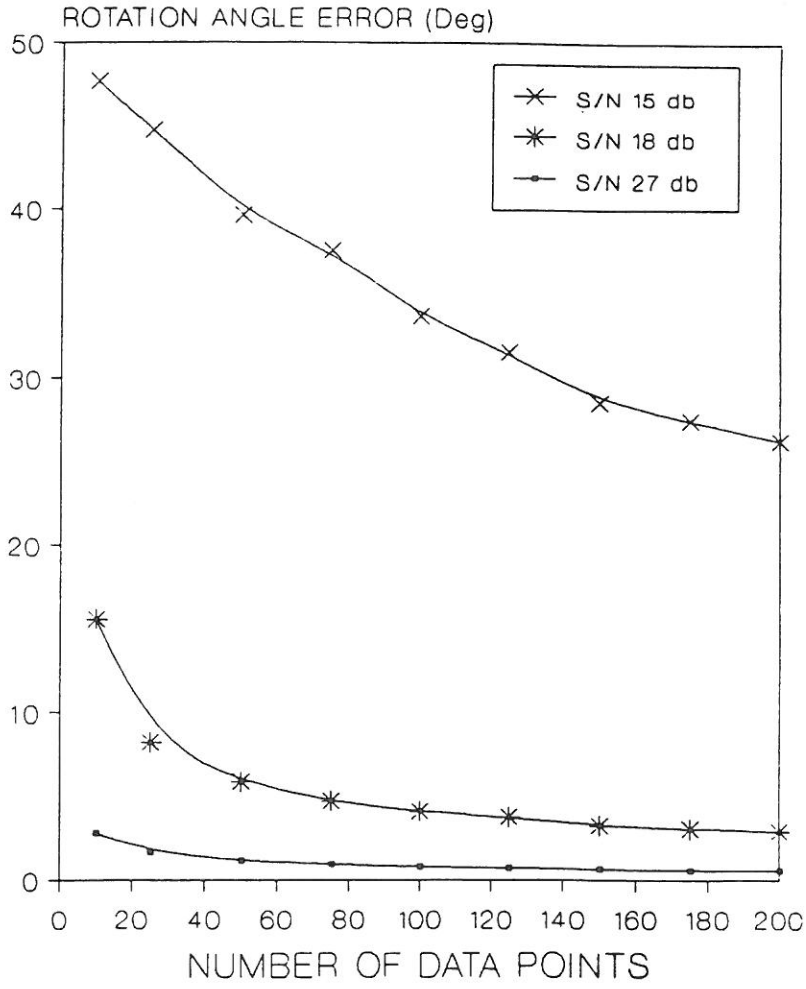

Figure 21: Mean rotation angle error versus number of points with uniform noise.

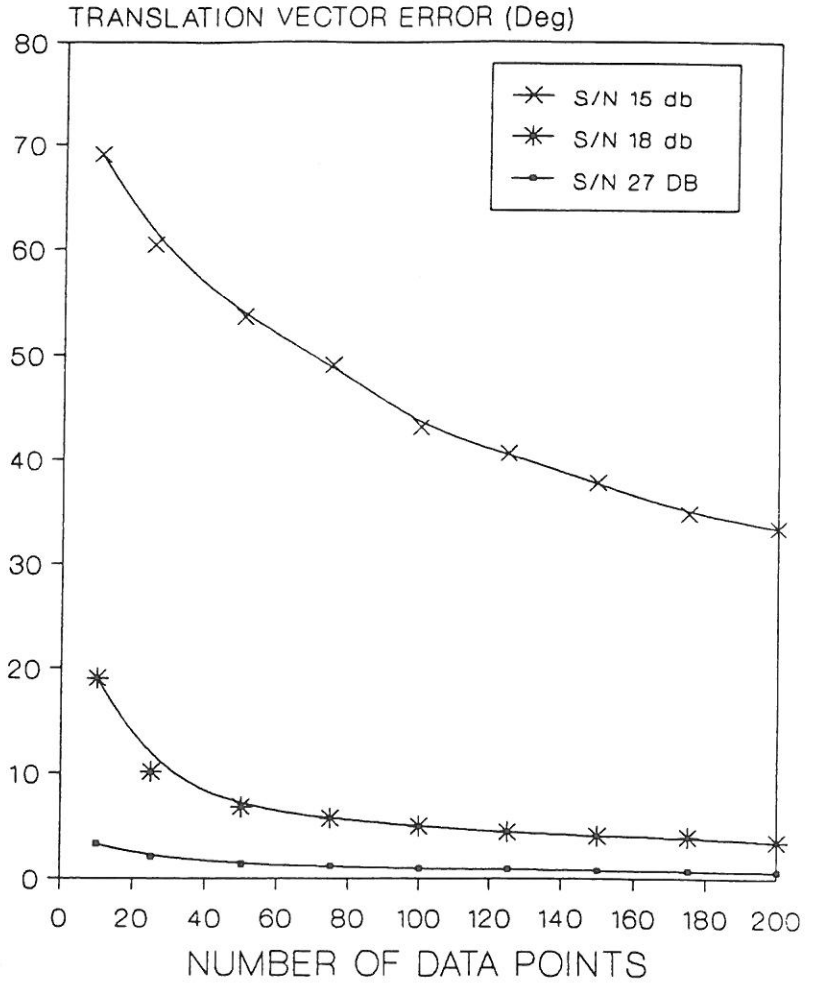

Figure 22: Mean translation angle error versus number of points with uniform noise.

4 2D Perspective Projection–3D Pose Estimation

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

$$\begin{aligned} u_{n1} &= f\frac{r_1 y_n + t_1}{r_3 y_n + t_3} \\ u_{n2} &= f\frac{r_2 y_n + t_2}{r_3 y_n + t_3} \\ t &= (t_1, t_2, t_3)' \\ R &= \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} \end{aligned} \tag{40}$$

where f, the focal length, is the distance of the image plane in front of the origin which is the center of perspectivity. In the 3D 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 3D model points and the corresponding 2D perspective projection points on the image plane. This problem is known as the exterior orientation problem in the photogrammetry literature. The dissertation by Szczepanski (1958) 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 (1981). Wrobel and Klemm (1984) discuss the fact that there are configurations of points for which the solution is unstable.

4.1 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 k^{th} 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.

Method 1

One iterative procedure for determining a least squares solution for R and t is

- (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 $d_n^k, n = 1, \ldots, N$ are given. Define the depth values for the $(k+1)^{th}$ iteration by:
  - (2.1) Find the rotation matrix R_k and the translation vector t_k which minimizes

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

    where the $\{w_n \mid n = 1, \ldots, N\}$ are non-negative weights reflecting the goodness of the observations. R_k and t_k constitute the solution to the 3D-3D pose estimation problem.
  - (2.2) Define $d_n^{k+1} = x_{n3}^k$, where

$$x_n^k = R_k y_n + t_k.$$

– (2.3) Define

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

where

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

and

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

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

A typical convergence characteristic of the computed depth values is shown in Figure 23. 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.

Method 2 Replace the step (2.2) of Method 1 with (2.4). Define d_n^{k+1} by

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

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

$$\begin{aligned}
\epsilon_{k+1}^2 &= \sum_{n=1}^{N} w_n \|R_{k+1} y_n + t_{k+1} - d_n^{k+1} v_n\|^2 \\
&\leq \sum_{n=1}^{N} w_n \|R_k y_n + t_k - d_n^{k+1} v_n\|^2 \\
&= \sum_{n=1}^{N} w_n \|(x_n^k - d_n^k v_n) + (d_n^k v_n - d_n^{k+1} v_n)\|^2 \\
&= \sum_{n=1}^{N} w_n [\|(x_n^k - d_n^k v_n)\|^2 + 2(x_n^k - d_n^k v_n)'(d_n^k - d_n^{k+1}) v_n \\
&\qquad + (d_n^k - d_n^{k+1})^2 \|v_n\|^2]
\end{aligned}$$

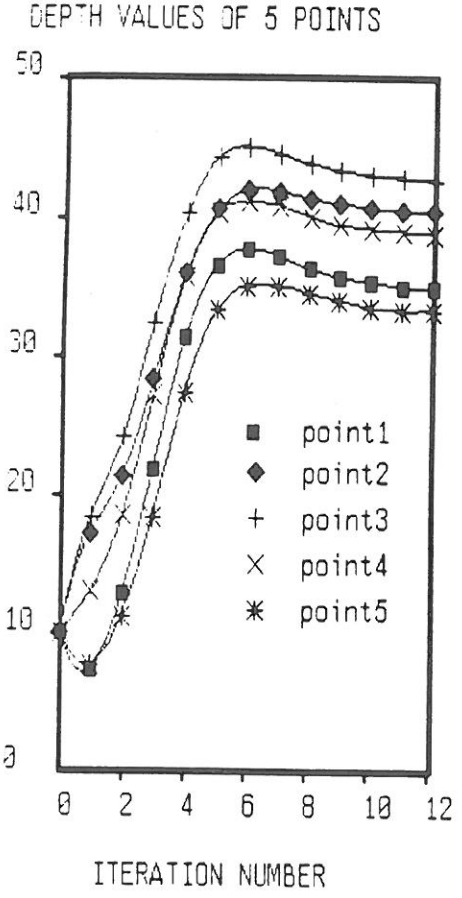

Figure 23: illustrates the convergence characteristics of Method 1. Convergence is achieved in about ten iterations.

$$
\begin{aligned}
&= \epsilon_k^2 + \sum_{n=1}^{N} w_n (d_n^k - d_n^{k+1}) \left[2(x_n^k - d_n^k v_n)' v_n + (d_n^k - d_n^{k+1}) \|v_n\|^2 \right] \\
&= \epsilon_k^2 + \sum_{n=1}^{N} w_n (d_n^k - d_n^{k+1}) [\ 2 x_n^{k\prime} v_n - 2 d_n^k \|v_n\|^2 \\
&\qquad\qquad + (d_n^k - d_n^{k+1}) \|v_n\|^2] \\
&= \epsilon_k^2 + \sum_{n=1}^{N} w_n (d_n^k - d_n^{k+1}) \left[2 x_n^{k\prime} v_n - (d_n^k - d_n^{k+1}) \|v_n\|^2 \right] \\
&= \epsilon_k^2 + \sum_{n=1}^{N} w_n \|v_n\|^2 \left[(d_n^{k+1})^2 - 2 \frac{x_n^{k\prime} v_n}{\|v_n\|^2} d_n^{k+1} + 2 \frac{x_n^{k\prime} v_n}{\|v_n\|^2} d_n^k - (d_n^k)^2 \right] \\
&= \epsilon_k^2 + \sum_{n=1}^{N} w_n \|v_n\|^2 \left[\left(d_n^{k+1} - \frac{x_n^{k\prime} v_n}{\|v_n\|^2} \right)^2 - \left(d_n^k - \frac{x_n^{k\prime} v_n}{\|v_n\|^2} \right)^2 \right] \qquad (46)
\end{aligned}
$$

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

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

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

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

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

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

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

$$\epsilon_{k+1}^2 \leq \epsilon_k^2 \qquad (50)$$

A typical convergence characteristic of the computed depth values is shown in Figure 24. 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.

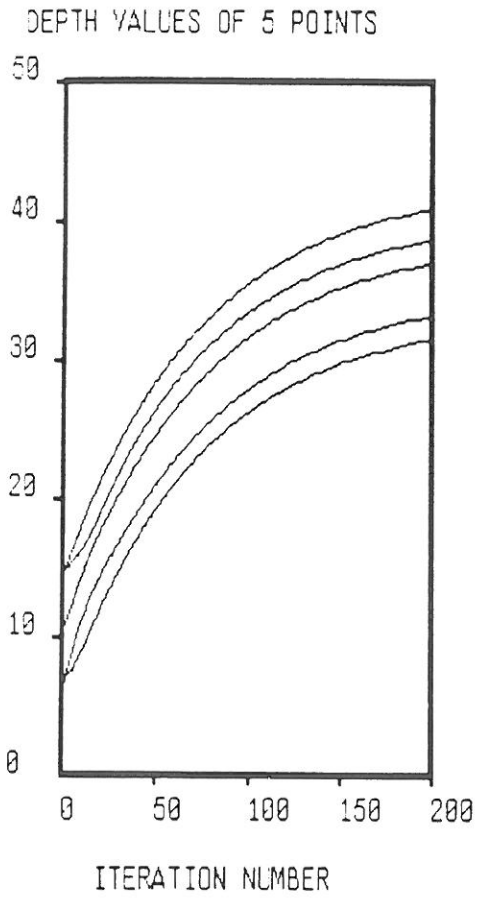

Figure 24: illustrates the convergence characteristics of Method 2. Convergence has been observed to be monotonic and is achieved in a few hundered iterations.

4.2 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

$$u_{ni} = u_{ni}^0 + \nu_{ni}, i = 1, 2, \quad n = 1, \ldots, N \tag{51}$$

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

$$\begin{aligned} \phi &= \phi^0 + \Delta\phi \\ \theta &= \theta^0 + \Delta\theta \\ \psi &= \psi^0 + \Delta\psi \\ t_i &= t_i^0 + \Delta t_i, i = 1, 2, 3 \end{aligned} \tag{52}$$

where $\phi^0, \theta^0, \psi^0, t_1^0, t_2^0$ and t_3^0 are some approximations, and $\Delta\phi$, $\Delta\theta$, $\Delta\psi$, Δt_1, Δt_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

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

$$F_{n2} = u_{n2} - f\frac{r_2 y_n + t_2}{r_3 y_n + t_3} \tag{54}$$

These equations can be linearized by Newton's first order approximation as follows:

$$\begin{aligned} F_{n1} &\simeq F_{n1}^0 + \nu_{n1} + b_{n11}\Delta\phi + b_{n12}\Delta\theta + b_{n13}\Delta\psi + \\ &\quad b_{n14}\Delta t_1 + b_{n15}\Delta\ t_2 + b_{n16}\Delta t_3 \\ F_{n2} &\simeq F_{n2}^0 + \nu_{n2} + b_{n21}\Delta\phi + b_{n22}\Delta\theta + b_{n23}\Delta\psi + \\ &\quad b_{n24}\Delta t_1 + b_{n25}\Delta\ t_2 + b_{n26}\Delta t_3 \end{aligned}$$

where

$$\begin{aligned} b_{ni1} &= \left(\tfrac{\partial F_{n1}}{\partial\phi}\right)^0, \quad b_{ni2} = \left(\tfrac{\partial F_{n1}}{\partial\theta}\right)^0 \\ b_{ni3} &= \left(\tfrac{\partial F_{n1}}{\partial\psi}\right)^0, \quad b_{ni4} = \left(\tfrac{\partial F_{n1}}{\partial t_1}\right)^0 \\ b_{ni5} &= \left(\tfrac{\partial F_{n1}}{\partial t_2}\right)^0, \quad b_{ni6} = \left(\tfrac{\partial F_{n1}}{\partial t_3}\right)^0 \end{aligned}$$

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

$$\begin{pmatrix} b_{111} & b_{112} & \cdots & b_{116} \\ b_{121} & b_{122} & \cdots & b_{126} \\ \vdots & \vdots & \ddots & \vdots \\ b_{N11} & b_{N12} & \cdots & b_{N16} \\ b_{N21} & b_{N22} & \cdots & b_{N26} \end{pmatrix} \begin{pmatrix} \Delta\phi \\ \Delta\theta \\ \Delta\psi \\ \Delta t_1 \\ \Delta t_2 \\ \Delta t_3 \end{pmatrix} = \begin{pmatrix} -F_{11}^0 \\ -F_{12}^0 \\ \vdots \\ -F_{N1}^0 \\ -F_{N2}^0 \end{pmatrix} - \begin{pmatrix} \nu_{11} \\ \nu_{12} \\ \vdots \\ \nu_{N1} \\ \nu_{N2} \end{pmatrix}$$

or simply

$$B\Delta = F - \nu \tag{55}$$

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 becomes negligibly small.

4.3 Robust M–Estimation

This section repeat some robust techniques used in nonlinear regression problems as mentioned in section 2. In particular, it can be used

to solve robustly the equation $B\Delta = F - \nu$ which results from the linearization of the original pose estimation problem. Any estimate T_k defined by a minimization problem of the form.

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

or by an implicit equation

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

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

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

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

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

where

$$w_i = \frac{\psi(x_i - T_k)}{x_i - T_k}, \quad i = 1, \ldots, n \tag{60}$$

This gives a formal representation of T_k as a weighted mean

$$T_k = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} \tag{61}$$

with weights depending on the sample (Huber, 1981). 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.

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

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

Tukey's ψ function can be expressed as

$$\psi(x) = \begin{cases} x\left(1 - \left(\frac{x}{a}\right)^2\right)^2, & \text{if } |x| \leq a; \\ 0, & \text{if } |x| > a. \end{cases} \tag{62}$$

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

$$f_i(\theta) = y_i, \quad i = 1, \ldots, n \tag{63}$$

which minimizes

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

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.

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

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, 1979).

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 above (Huber, 1981).

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

- (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.
  - (2.1) Compute the modified residuals r_i^* for $i = 1, \ldots, n$.

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

where

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

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

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

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

The solution for this equation can be found using the standard least squares method. If the singular value decomposition of the matrix X is $X = U_1 \Sigma_1 V'$, then the solution is $\hat{\delta} = V\Sigma_1^{-1} U_1' r^*$.

  - (2.3) Set $\theta^{k+1} = \theta^k + \hat{\delta}$.

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

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

In the standard weighted form

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

where

$$w_i = \frac{\psi(\frac{r_i}{S})}{(\frac{r_i}{S})} \tag{68}$$

Therefore, the iterative procedure to determine θ is

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

$$PX\delta = Pr$$

where

$$P = \begin{pmatrix} \sqrt{w}_1 & & \\ & \ddots & \\ & & \sqrt{w}_N \end{pmatrix}$$

  - (2.2) If $\hat{\delta}$ is the solution in step (2.1), then set

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

4.4 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 signal to noise ratio (SNR), the number of matched points, or the number of outliers, which will be defined later.

Data Set Generation A set of 3D model points

$$y_i = (y_{i1}, y_{i2}, y_{i3})', i = 1, \ldots, N$$

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 3D model points are rotated and translated in the 3D space forming a set of 3D points $x_i, i = 1, \ldots, N$. 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 3D points, x_i, are replaced with randomly generated 3D points, $z_i = (z_{i1}, z_{i2}, z_{i3})', i = 1, \ldots, M$. M is the number of the replaced 3D points and

$$\begin{aligned} z_{i1} &= t_1 + \nu_{i1} \\ z_{i2} &= t_2 + \nu_{i2} \\ z_{i3} &= x_{i3} \end{aligned} \tag{69}$$

where $\nu_{i1}, \nu_{i2}, i = 1, \ldots, M$ 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 2D points, $x_i, i = 1, \ldots, N$ are perspectively projected onto the image plane. Given the 3D model points and the corresponding 2D 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 3D 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

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

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

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 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 (20db to 80db in 10db step) for different PO (0% to 20% in 5% step).
- E2: Set SNR = 40db. Estimate the sum of three rotation angle error against PO (0% to 20% in 5% step) for different N (10 to 50 by steps of 10).
- E3: Set PO = 10%. Estimate the sum of three rotation angle error against SNR (20db to 80db in 10db step) for different N (10 to 50 by steps of 10).

Figure 25 shows the results of E1, E2, and E3 performed for the initial approximation algorithm using iterative least squares solution (A1), method 2 of section 4.1.1.2. 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. Figures 26 and 27 show the result of the least squares adjustment by linearization algorithm (A2), algorithm in section 4.1.2, and the robust M-estimate algorithm (A3), modified weights algorithm in section 4.5.2, respectively. Figure 28 compares the three algorithms A1, A2, and A3 in the experiment set E1. Figures 29 and 30 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%, algorithms A2 and A3 are applied for SNR from 20db to 40db in a step of 10db, and the algorithm A2 is applied for N = 18, PO = 0% and SNR from 20db to 40db in a step of 10db. This compares the efficiency of the robust technique against the non–robust technique in the case where

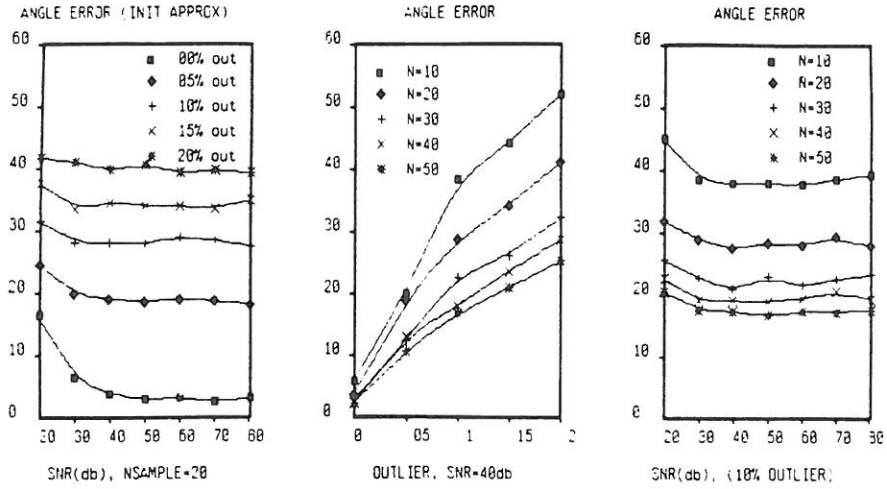

Figure 25: illustrates the performance characteristics for the initial approximation solution (Method 2).

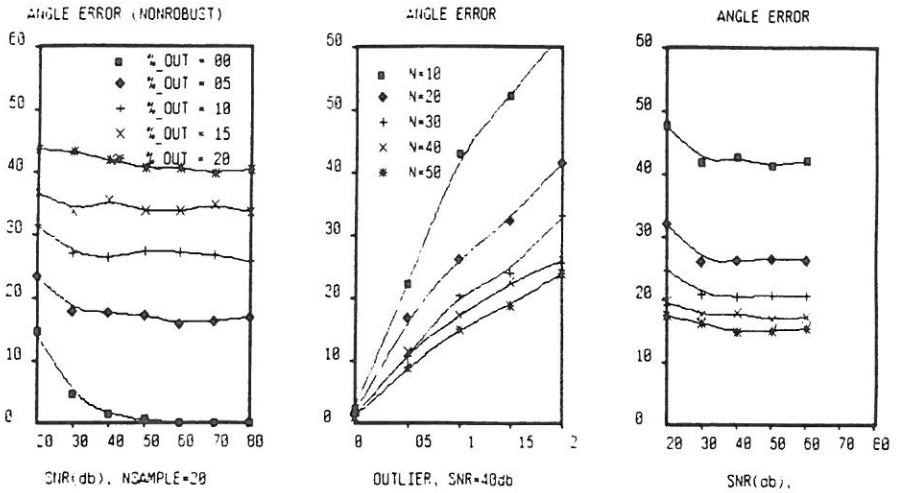

Figure 26: illustrates the performance characteristics of the least squares adjust by linearization.

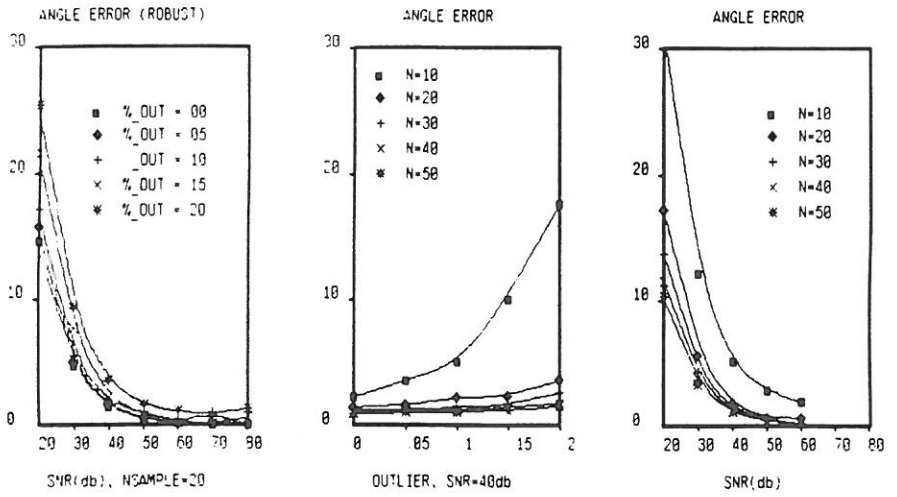

Figure 27: illustrates the performance characteristics of the robust M–estimate algorithm.

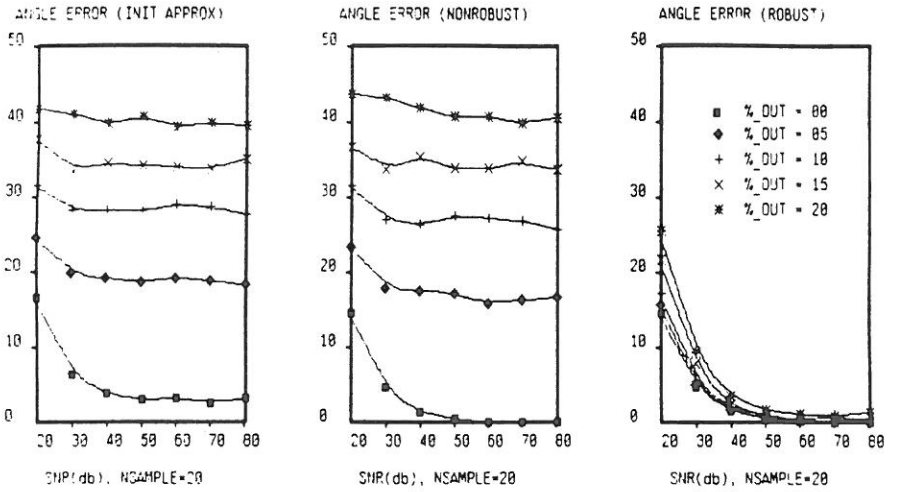

Figure 28: illustrates the performance charcteristics of angle error as a function of signal to noise ratio for the initial approximation method, the non–robust linearized least squares adjustment, and the robuts M–estimate.

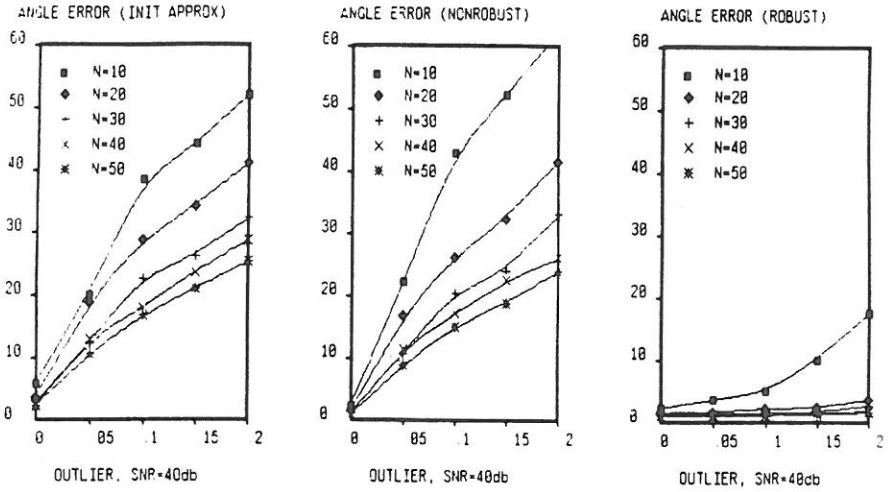

Figure 29: illustrates the performance characteristics of angle error versus fraction of outliers for the initial approximation method, the linearized least squares adjustment, and the robust M–estimate.

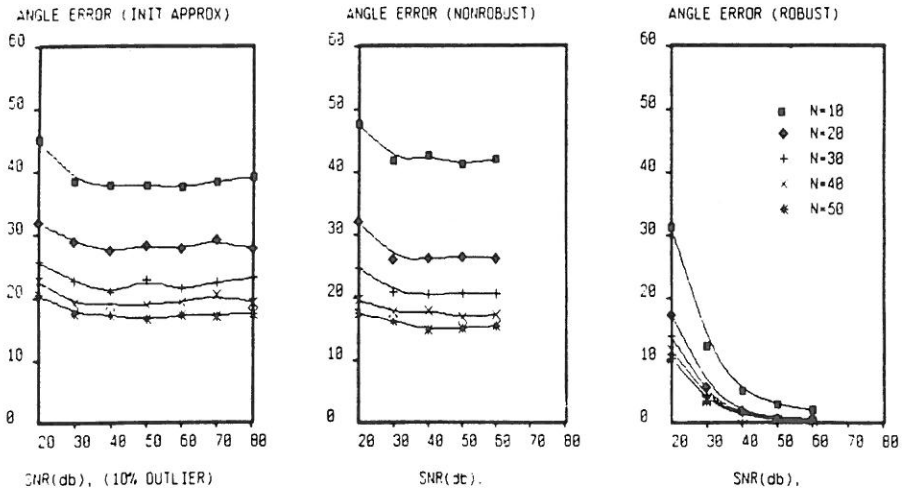

Figure 30: illustrates the performance characteristics of angle error versus fraction of outliers for the initial approximation method, the linearized least squares adjustment, and the robust M–estimate.

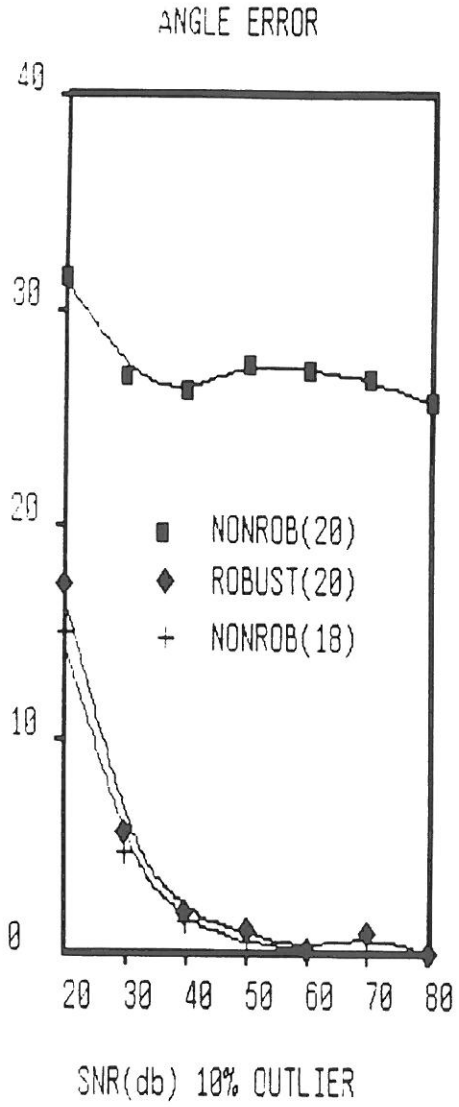

Figure 31: illustrates the efficiency of the robust technique operating on a data set of 20 points, 18 points having Gaussian noise and 2 outliers, against the non–robust technique operating on a data set having 18 points having Gaussian noise.

the non–robust technique uses only the non–outlier points given to the robust technique. Figure 31 shows the result of this experiment.

5 2D Perspective – 2D 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 3D parameter estimation, but few of these contributions systematically discuss the effect of noise. Thompson (1959) developed the nonlinear equations using the form resulting from the correspondence of 2D perspective projection points on one image with 2D perspective projection points on another image. He gave a solution which determines a rotation matrix guaranteed to orthonormal. His method was to linearize the non–linear equations and iterate. Roach and Aggarwal (1980) 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(1981) , extended by Tsai and Huang(1984), unified by Xinhua Zhung, T.S. Huang, and R. M. Haralick(1986), and simplified by Xinhua Zhung 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 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 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 and Haralick (1986) 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.

5.1 Simplified Linear Algorithm

As shown in Fig. 32 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 $z \leq 0$. Let $P = (x, y, z)^t$ represent a set of object points coordinate before motion and $P' = (x', y', z')^t$ represent the same set of object points coordinate after motion. The point coordinate $[x_i, y_i, z_i] \in p$ is corresponding to $[x_i', y_i', z_i'] \in P'$. Let (X,Y), (X', Y') represent the perspective coordinate of P and P' onto the image plane z=1. These give

$$\begin{aligned} X &= x/z \\ Y &= y/z \\ X' &= x'/z' \\ Y' &= y'/z'. \end{aligned} \tag{72}$$

The rigid body motion equation is given as follows:

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

where R_o is an 3 x 3 rotation matrix (orthonormal); T_o is 3 x 1 translation vector. In terms of Euler angles ψ, θ, and ϕ the rotation

matrix can be represented as follows:

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

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

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 Eq. 73 we obtain

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

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

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

where T x $R_o = [T \times r_1, T \times r_2, T \times r_3]$, 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 homogenous equations w. r. t. nine elements of E:

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

Relation (77) was originally shown by Thompson(1959). Suppose that we have N correspondences. 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 the Eq. 77 can be transformed into the overconstraint linear equation for h

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

Solving Eq. 78 in the least squares sense we seek an estimator h which minimizes $\|Ah\|^2$. The 9 component vector h is found to be the right eigenvector of A having smallest singular value. Any $T \times R_0$ with $T \times T_0 = 0$ satisfies Eq. 77. Moreover, such a colinear vector T has one degree of freedom when $T_0 \neq 0$ or three degrees of freedom when $T_0 = 0$. Thus the general solution of the Two–View Motion Eq. 77 has at least one degree of freedom when $T_0 \neq 0$ or three degrees of freedom when $T_0 = 0$.

When $T_0 \neq 0$, the nine elements of E must have a rank 8, and $T_0 = 0$ the nine elements of E must have rank 6. Under the surface assumption (Zhuang, Haralick, and Huang, 1986) the number of image corresponding point pairs must be at least 8 when $T_0 \neq 0$, or greater than or equal to 6 when $T_0 = 0$. 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 3D points corresponding to the observed image points do not lie on a quadratic surface passing through O and O' when $T_0 \neq 0$ or a cone with its apex at O when $T_0 = 0$.

Decomposing E

E has two decompositions; $T \times R_0$ and $(-T) \times R_0$ with R_0 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 2D space and also $\|E\| = \sqrt{2}\|T\|$. Therefore we can get three constraints as follows:

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

$$\|E\| = 2\|T\| \tag{79}$$
$$E^t T = 0$$

We can use the least square method to solve Eq. 79 for T and obtain the value of the T vector from the other two constraints. Since T is colinear with T_0, T_0 should have the same orientation as T or $-T$. Taking a cross-product with both sides of Eq. 73 by $(X', Y', 1)^t$ we obtain

$$z_1(X', Y', 1)^t \times [R_0(X, Y, 1)^t] + (X', Y', 1)^t \times T_0 = 0. \tag{80}$$

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

$$\sum_{i=1}^{n} (X_i', Y_i', 1)^t \times [R_0(X_i, Y_i, 1)^t](X_i', Y_i', 1)^t \times T \geq 0 or \leq 0. \tag{81}$$

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 $E = [E_1, E_2, E_3]$

5.2 The Robust Algorithm

As mentioned in the previous section, Eq. 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 bi-weight. The difference between the biweight estimator and the least-squares estimator is briefly discussed.

Biweight Estimator

Let x_i be the i^{th} 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 w.r.t. $\hat{x}$ 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 useful data. On the other hand, when the value is large the outliers can not 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 $\psi(u) = \mathrm{w(u)u}$. Thus, the weight funtion $w(u)$ can be represented

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

Robust Estimation of E

From above equation we can see that the biweight estimator is a weighted least square estimator. With the weight matrix we rewrite Eq. 78:

$$WAh = 0. \tag{87}$$

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

$$WA = U \sum V^t. \tag{88}$$

where

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

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

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

The index n is 9 and m is the number of corresponding point pairs. The eigenvector of V which 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(1981). Hence, the residual errors need to be adjusted according to the following

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

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

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

We can simplify the above equation by substituting $U \sum V^t$ for WA. After some linear algebra manipulation Eq. 90 becomes

$$H = U_a U_a^t. \tag{91}$$

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

It is trivial to then obtain $h_{ii} = \sum_{k=1}^{9} u_{ik}^2$. Once h_{ii} are obtained, then they can be substituted into Eq. 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.

5.3 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. 32, the image frame is located at $z = 1$. By mapping 3D spatial coordinates into image frame, and then adding noise to the points before and after motion, we obtain

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

Signal is related to object image size, and noise may come from camera error, digitization, or corresponding point extraction error. Define SNR $= 20 \log \frac{signal}{\sigma}$ db, where σ is the standard deviation. In the simulation experiments , the 3-D spatial coordinates before motion (x,y,z), true 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 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 or uniform 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 absulute error. For each experimental condition a thousand trials are done. Mismatching noise is simulated by randomly swapping one componenet 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 4 steps. The results are shown in Fig's. 33–36. When noise-free, the linear algorithm has excellent performance with zero error for all cases. Figures 33–36 show the translation error and rotation degree error, which can define an average of mean alsulute error of three Euler angles, versus the signal to noise ratio for different numbers of corresponding point pairs for both Gaussian noise and uniform noise. It shows that the error increases as the noise level increases. Furthermore, depending on kind of noise and number of corresponding point pairs, the error increases very rapidly when the signal to noise ratio gets below a knee value. Table 1 shows the minimum signal to noise ratio 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. Fig.37 a.b.c.d. shows 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 Fig.38 a.b.c.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 Fig.39 a.b.c.d. They show results similar to the outlier results. Fig.40 shows the standard deviation of the points plotted in Fig.37, Fig.38, Fig.39. The behaviors of the standard deviation of the three rotation angles are similar, hence we put them together and take average.

5.4 Summary of Robust algorithm

- Step 0. Use the identity matrix for initial weight matrix.
- Step 1. Use singular value decomposition to solve Eq. 87
- Step 2. Update the weight matrix by Eq. 86.

  Repeat Step 1. and 2. until the criteria satisfied.
- Step 3. Determine the translation vector from Eq. 79 and Eq. 81
- Step 4. Obtain true R_o from Eq. 82

6 Conclusion

The noise behavior for the general linear motion algorithm and its robust version was determined from over hundred thousand experimental trials. The experimental results indicated that the robust algorithm can extract the 3-D motion parameters with one degree rotation mean absolute error from image sequences which contain 30 percent of outliers, this is much better than the linear algorithm which has more than ten degree rotation mean absolute error. The robust algorithm can detect the ouliers, mismatching errors, and

blunders. Therefore, it can be an effective tool in estimating 3-D motion parameters from multiframe time sequence imagery. It should prove equally effective when applied to image flow data.

	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

Table 1. SNR (db) for mean absulute error in 1 degree.

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.

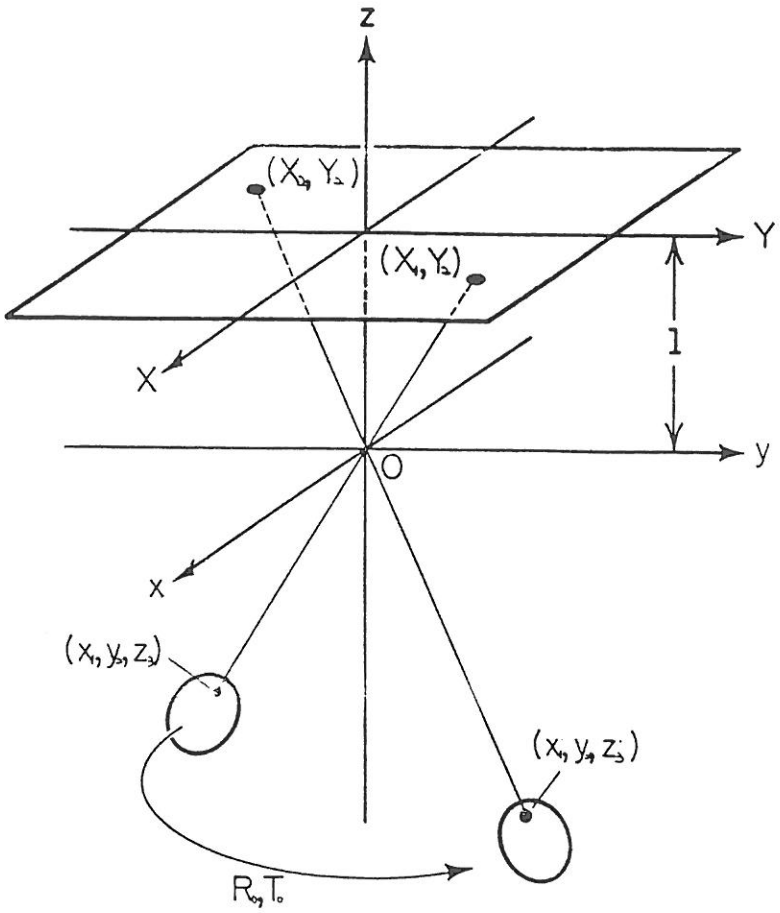

Figure 32: Imaging Geometry

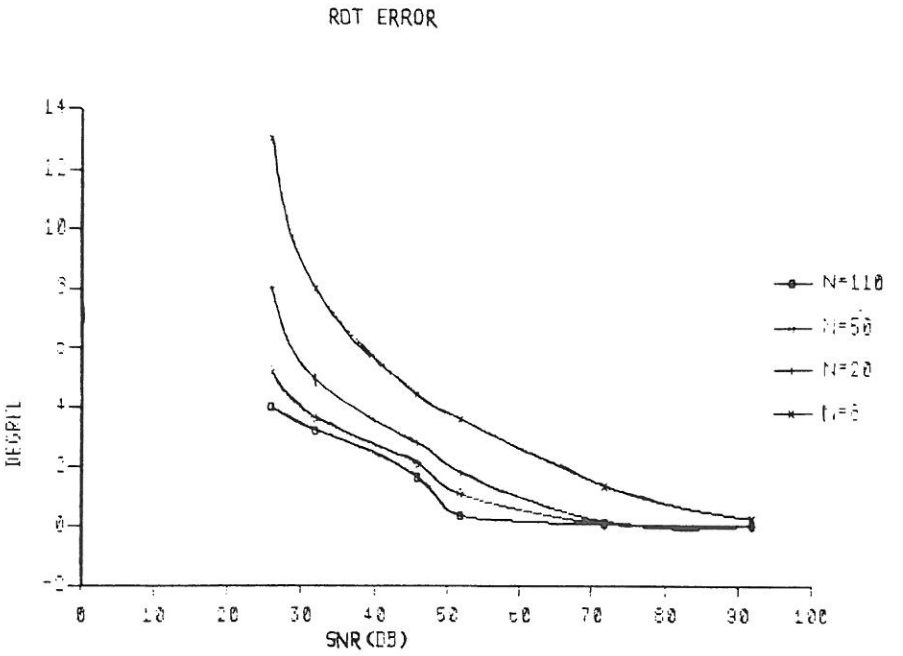

Figure 33: Mean angle error between the estimated rotation angles and the true rotation angles versus the Gaussian noise level for four corresponding point data set sizes of 8 to 110 pairs. Each point on the graph represents 1,000 trials.

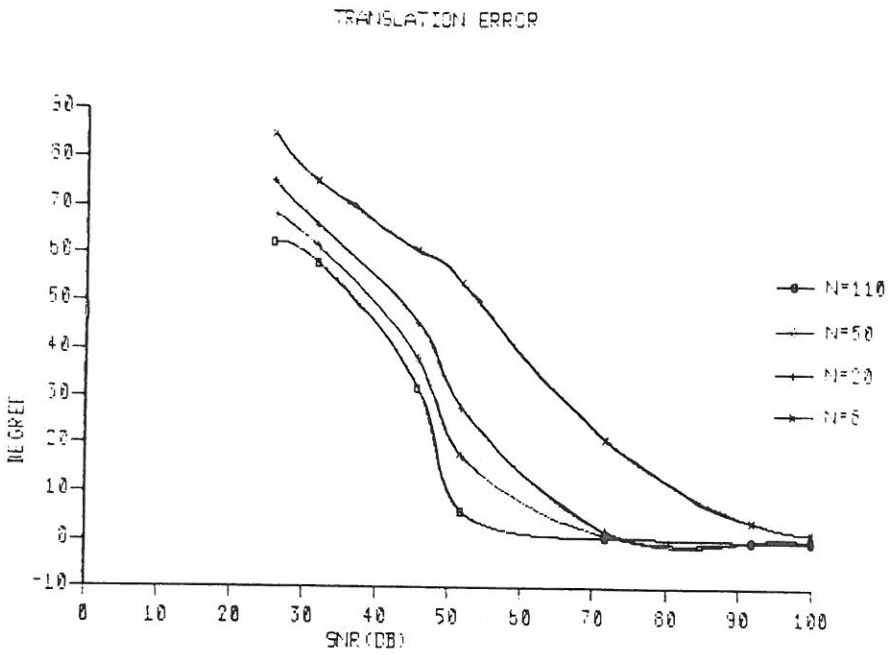

Figure 34: Mean angle error between the calculated translation vector and the true translation vector versus the Gaussian noise level for four corresponding point data set sizes of 8 to 110 pairs. Each point on the graph represents 1,000 trials.

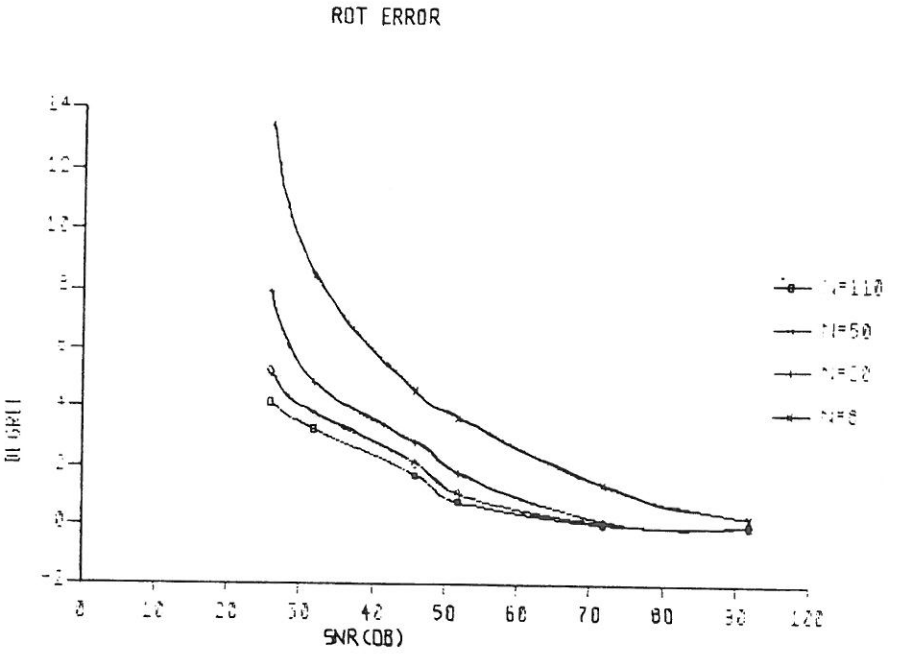

Figure 35: Mean angle error between the estimated rotation angles and the true rotation angles versus the uniform noise level for four corresponding data set sizes 8 to 110 pairs. Each point on the graph represents 1,000 trials.

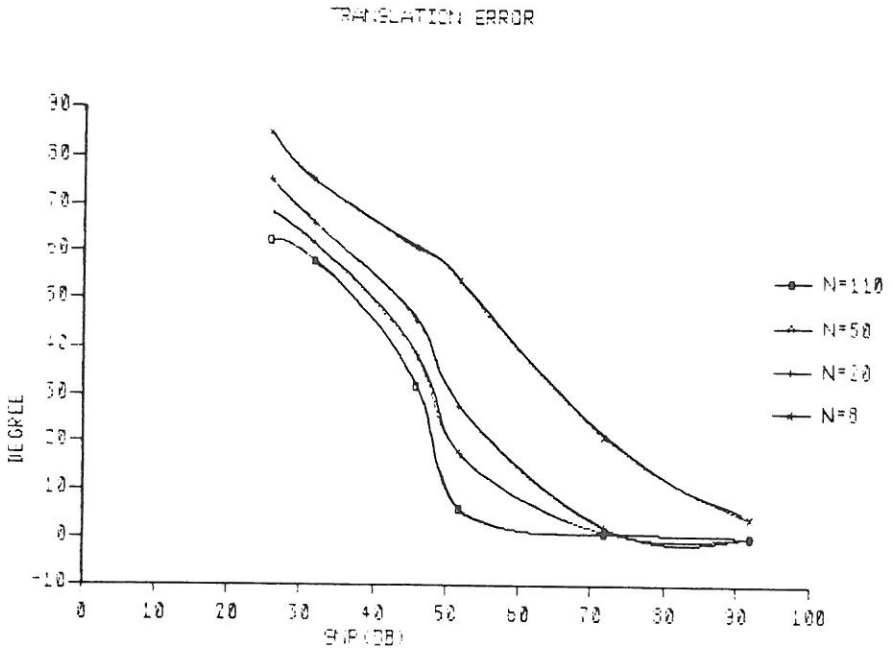

Figure 36: Mean angle error between the estimated translation vector and the true translation vector versus the uniform noise level for four corresponding point data set sizes of 8 to 110 pairs. Each point on the graph represents 1,000 trials.

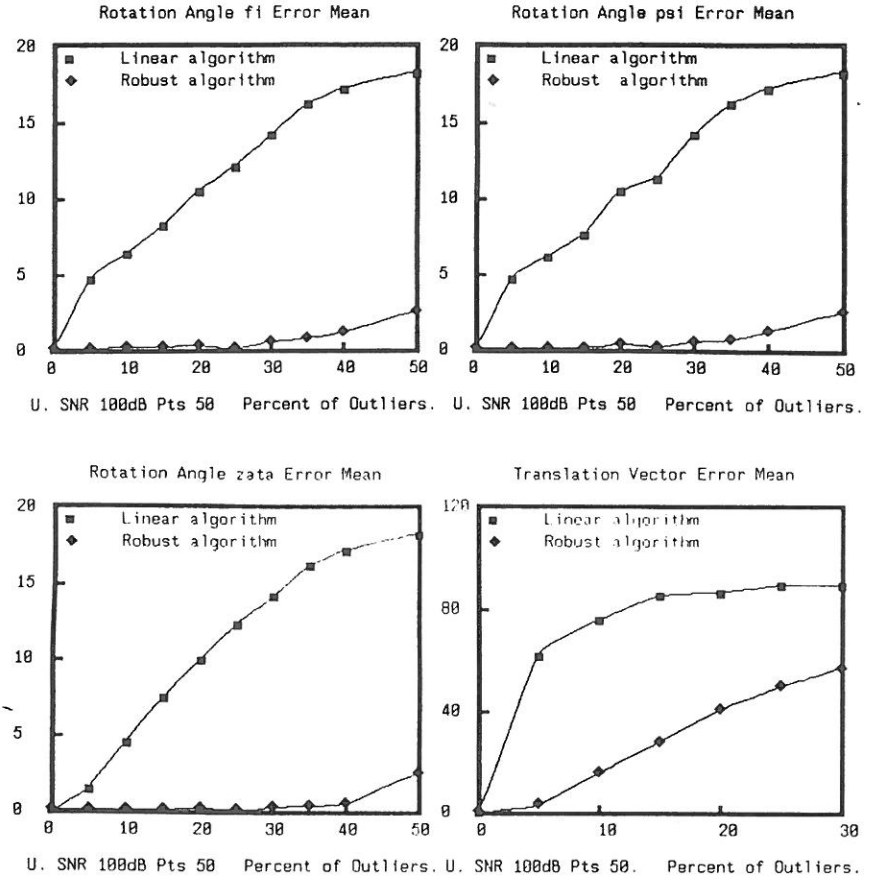

Figure 37: a.b.c.d. Compares the ϕ, ψ, θ angle error and translation angle error between the linear algorithm and robust algorithm for different percent of outliers. The noise is uniform with 100dB SNR. The number of points is 50. Each point on the graph represents 1,000 trials.

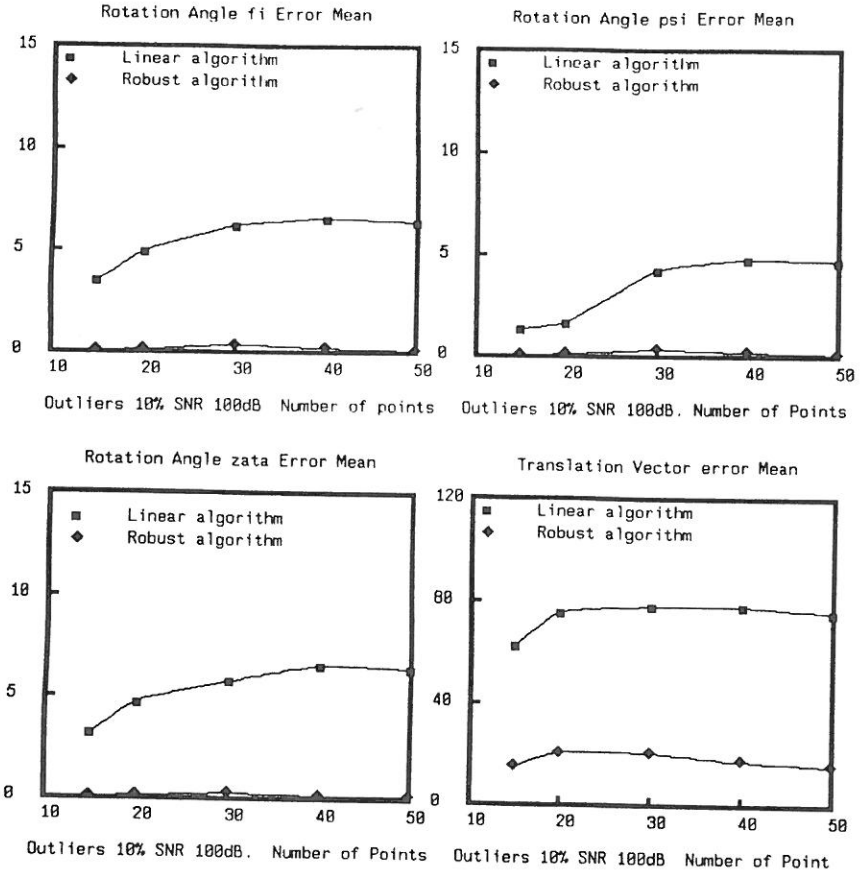

Figure 38: a.b.c.d. Compares the ϕ, ψ, θ angle error and translation angle error between the linear algorithm and robust algorithm for different number of points. The noise is uniform with 100dB SNR . The percent of outliers is 10 %. Each point on the graph represents 1,000 trials.

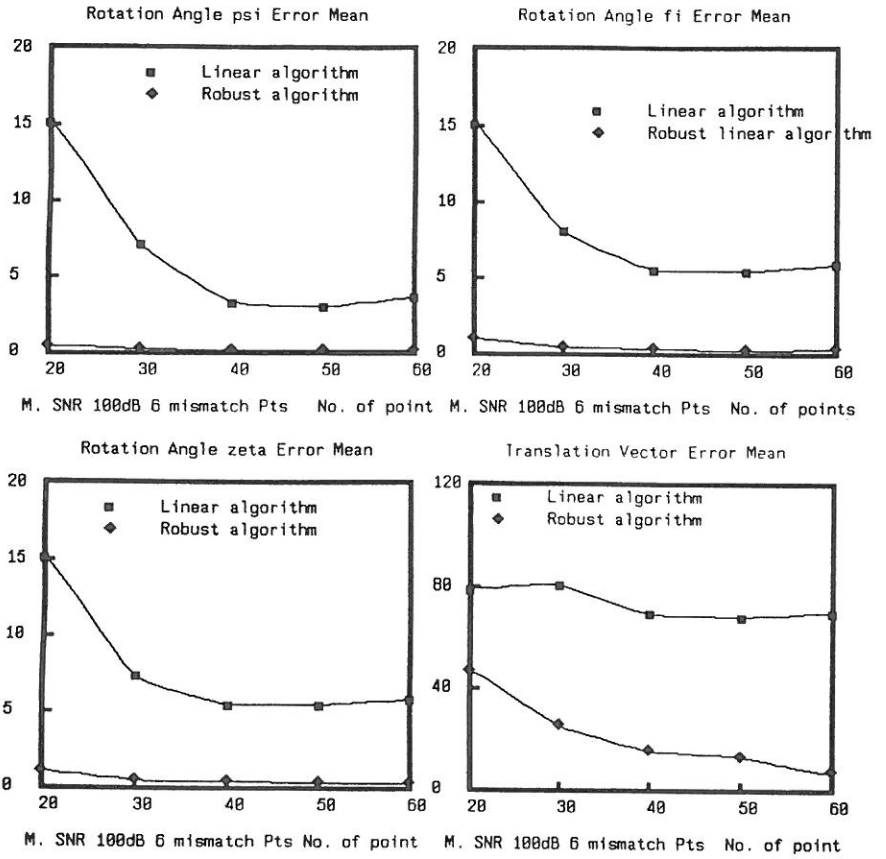

Figure 39: a.b.c.d. Compares the ϕ, ψ, θ angle error and translation angle error between the linear algorithm and robust algorithm for different number of points. The noise is uniform with 100dB SNR and is added six points of mismatch. Each point on the graph represents 1,000 trials.

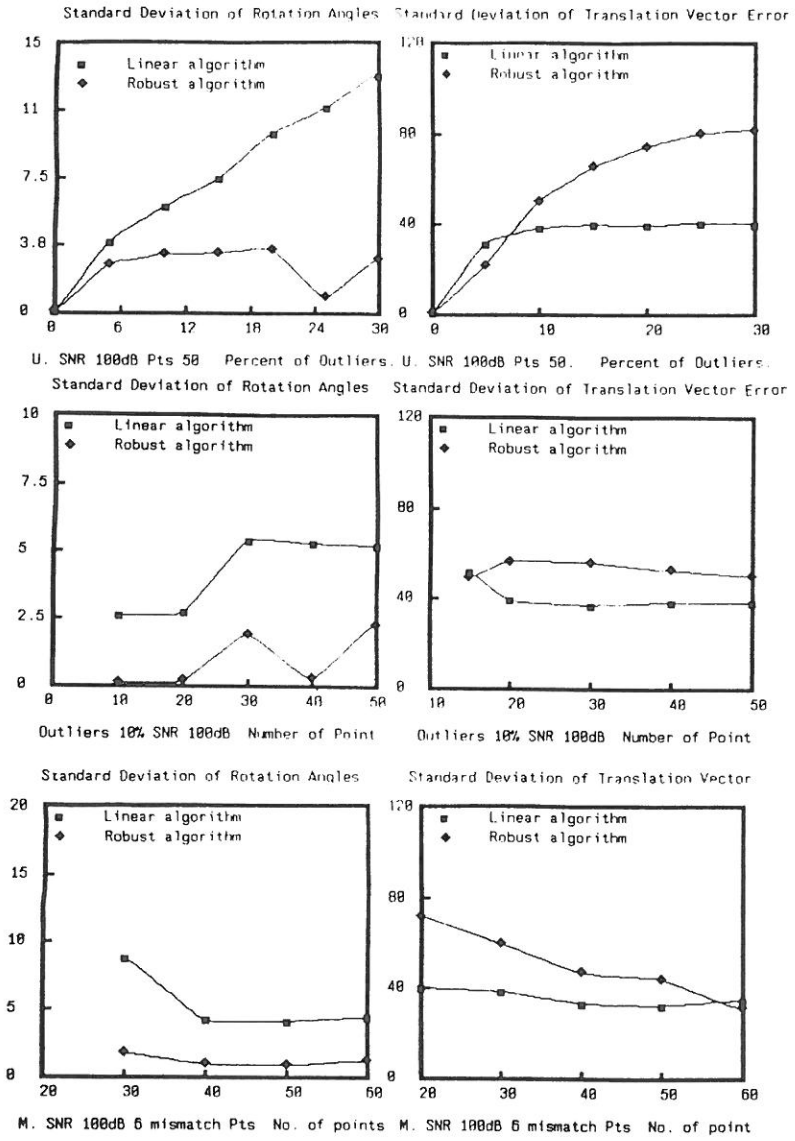

Figure 40: a.b.c.d.e.f. The standard deviation of the points plotted in Fig.37, Fig.38, Fig.39

Bibliography

[1] Blais J.A.R. *Three-Dimensional Similarity,* The Canadian surveyor, No.1 1972, pp. 71-76

[2] C. Longuet - Higgins, *A Computer Algorithm for Reconstructuring a scence from Two Projectives,* Nature 293, 133 - 135(1981)

[3] Drehungen, International Archives of Photogrammetry and Remote Sensing, Vol. 25, Part A3b, pp. 1153-1163.

[4] Fang, J.Q., and T.S. Huang, *Some experimentson estimating the 3D motion parameters of a rigid body from two consecutive image frames,* IEEE Trans. Pattern Analy. Mach. Intell., PAMI-6, 1984, pp. 547-554.

[5] Fernando Sanso *An Exact Solution of The Roto-Translation Problem* Photogrammetria, 29(1973) pp. 203-216

[6] Granshaw S.I. *Relative Orientation Problems,* Photogrammetric Record, 9(53) 1979, pp.669-674

[7] Hoaglin, C.F. Mosteller, and J.W.Tukey, *Understanding Robust and Exploratory Data Analysis,* Johy Wiley & Sons, Inc., 1983 pp. 348-349.

[8] Pope, J.A.,1970 An Advantageous, Alteernative Parametrization of Rotations for Analytical Photogrammetry. ESSA Tech. Rep., C and GS 39.

[9] Peter J. Huber, *Robust Statistics,* John Wiley & Sons, 1981.

[10] Robert V. Hogg,*An Introduction to Robust Estimation,* Robustness in Statistics edited by R.L. Launer & G.N. Wilkinson, Academic press, 1979.

[11] Roach, J.W., and J.K. Aggarwal, *Determining the movement of objects from a sequence of images,* IEEE Trans. Pattern Analy. Mach. Intell., PAMI-6, 1980, pp. 554-562.

[12] Schut, G.H., 1960 *On Exact Linear Equations for the Computation of Rotational Element of Absolute Orientation,* Photogrammetria, 15(1): pp. 34-37

[13] Shafer, Steven A. and Takeo Kanade, *Gradient Space under Orthography and Perspective,* Computer Vision, Graphics, and Image Processing, Vol. 24, 1983, pp. 182-199.

[14] Shafer, Steven A. and Takeo Kanade, *Using Shadows in Finding Surface Orientations,* Department of Computer Science, Carnegie–Mellon University, Pittsburgh, PA, 1982, pp. 1-61.

[15] Szczepanski, W., *Die Lösungsvorschläge für den räumlichen Rückwärtseinschnitt,* Deutsche Geodätische Kommission, Reihe C: Dissertationen—Heft Nr. 29, 1958, 1-144. Wrobel, B., and D. Klemm, "Über der Berechnung allgemeiner räumlicher

[16] Thompson,E.H. *An Exact Linear Solution of the Problem of Absolute Orientation,* Photogrammetria, XV, 4, 1958, pp. 163-178.

[17] Thompson,E.H. *A Rational Algebraic Formulation of The Problem of Relative Orientation* , The Photogrammetric Record, Vol.III No.14 1959 pp. 152-159

[18] Tienstrra J.M. *Calculation of Orthogonal Matrices,* ITC Delft Series, A48 1969

[19] Tsai, R.Y., and T.S. Huang, *Uniqueness and estimation of 3D motion parameters of rigid objects with curved surfaces,* IEEE Trans. Pattern Analy. Mach. Intell., PAMI-6, 1984, pp. 13-17.

[20] Wolfgang Forstner, *The Reliability of Block Triangulation. Photogrammetric Engineering and Remote Sensing,* Vol. 51, No. 6, August 1985, pp. 1137-1149.

[21] Zhuang, X., Haralick, R., and T.S. Huang, *Two-view motion analysis: a Unified Algorithm,* Opt. Soc. Am., Vol.3, No.9, 1986, pp. 1492-1450.

[22] Zhuang, X. and R. Haralick, *A Simplification Linear Algorithm,* to be published.