Robust Estimation for Motion Parameters

Changming Sun *†
A. K. Forrest

Centre for Robotics and Automated Systems
Imperial College of Science, Technology and Medicine
London SW7 2BX, U.K.

Abstract

The performance of least squares method can be improved by changing the error metric so that points which lie far from the bulk of data do not influence the final value—that is to reject the outliers. In this paper, the combination of two robust estimators are used to obtain motion parameters of a camera from matched image features. Results obtained show that the robust estimator has the ability to remove gross errors and mismatch points automatically. Therefore the existence of a small amount of outliers or mismatches will not affect the final results. Different robust estimators can be used for the purpose of parameters estimation. In our case the Huber and Tukey's estimators are used which allows ten per cent mismatches. Median absolute estimator can also be used which can allow as high as fifty per cent outliers in the whole corresponding points. But one of the disadvantage is that it will take much more time.

1 Introduction

The estimation of the three-dimensional motion parameters of a rigid body is an important problem in image analysis. An algorithm, in order to serve robotics applications (or explain visual abilities) must be robust against noise—a small percentage of noise in the input should not create catastrophic results in the output. A conventional least squares method attempts to minimize an error metric function in which the value at a particular parameter position is influenced by all feature points. The least squares estimator is linear and unbounded with an increase in any of the observed values. Outliers can easily destroy a least squares fit. Therefore it is necessary to identify and eliminate them. In typical estimation problems, a straightforward least squares method is not particularly useful. The linear motion estimation algorithm was developed by Longuet-Higgins [1], extended by Tsai and Huang [2]. The linear algorithm has an advantage of being

*Now at: CSIRO Division of Mathematics and Statistics, Locked Bag 17, North Ryde, NSW 2113, Australia.

†The authors would like to thank K. L. Chan for providing some of the test data.

BMVC 1994 doi:10.5244/C.8.73

simple and fast over the nonlinear algorithm as in [3]. It can also find a unique solution except in degenerate cases [2, 4]. The linear algorithm works well when there is limited noise and no matching error. However, it is highly sensitive to noise and mismatches. Lee *et al* [5] show that when real world image data produced by a vision system were used, a disaster occurs. Because of the nature of the linear algorithm, increasing the number of corresponding point pairs can only to some extent suppress the noise effect. Besl *et al* [6] show one way to apply the theory of robust statistics to the data smoothing and derivative estimation problems. They demonstrated a robust window operator that preserves grey level and gradient discontinuities in digital images as it smooths and estimates derivatives. It is believed that many computer vision algorithms could be substantially improved by using robust statistical procedures, especially algorithms involving multi-variate regression. Jones *et al* [7] investigate the problem of circle center estimation using robust, redescending estimators. In this paper we use two robust estimators to obtain an essential matrix which is later used for the calculation of motion parameters and the 3-D structures.

2 The Linear Equation

Let $\mathbf{X}' = (X', Y', Z')^T$, $\mathbf{X} = (X, Y, Z)^T$ be the three-dimensional coordinates of a point with respects to two viewpoints. They are connected by an arbitrary displacement, we may write

$$\mathbf{X}' = \mathbf{R}(\mathbf{X} - \mathbf{T}) \tag{1}$$

A general relationship between the two sets of image coordinates can be established. Define a new matrix $\mathbf{Q}$ as in [1]

$$\mathbf{Q} = \mathbf{R}\mathbf{S} \tag{2}$$

where $\mathbf{R}$ is a rotation matrix, and $\mathbf{S}$ is a skew-symmetric matrix with elements from the translation vector $\mathbf{T}$:.

$$\mathbf{S} = \begin{pmatrix} 0 & T_z & -T_y \\ -T_z & 0 & T_x \\ T_y & -T_x & 0 \end{pmatrix} \tag{3}$$

The following equation can be derived which describes the relationship between the image coordinates [1]:

$$\mathbf{x}'^T \mathbf{Q} \mathbf{x} = 0 \tag{4}$$

where $\mathbf{x}' = (x', y', 1)^T$, $\mathbf{x} = (x, y, 1)^T$, are matched image points.

Now we come to the step to determine the nine elements of $\mathbf{Q}$. There will be one equation for every matched point, so the ratios of the nine unknowns of $\mathbf{Q}$ can be obtained, in general, by solving eight (or if more than eight, by least squares fitting) simultaneous linear equations of type (4).

When solving the homogeneous equation (4), where $\mathbf{A}$ is an $N \times M$ matrix, if $N < 9$, the equation will has infinite number of solutions; if $N = 9$, when $rank(\mathbf{A}) = 9$, Eq (4) has only one zero solution; otherwise it has multiple solution; if $N > 9$, the least squares method is usually used, and SVD is employed to be able to handle the singular case. After the $\mathbf{Q}$ matrix has been obtained, the rotation matrix, translation vector, and the 3-D results can be obtained as in [1].

3 Robust Estimation

As in the case of the least squares method, robust estimation requires the minimization of an error criterion. This function F is composed of terms from each of the N data points. It is given by

$$F = \sum_{i=0}^{N-1} \rho(\epsilon_i) \tag{5}$$

where $\rho(\epsilon_i)$ is a function defining the effect of errors. The error term ϵ_i is a function of the i^{th} data points. In much of the robust regression, the numerical values of the estimates can be obtained only after an iterative process because the estimators do not have closed forms like the mean and variance. Instead, they come equipped with an algorithm for producing the value of the estimate—for example, by minimizing a function [8]. Modern research on robust methods offers even better performance if we can accept more complicated estimators, for instance, the M-estimators, the class of estimators that offers the advantages in performance, flexibility, and convenience. Table 1 lists some of the M-estimators.

Table 1: *M-Estimators for regression (t is the residual of an equation).*

Estimators	$\rho(t)$	$\psi(t)$	$w(t)$	range of t
Least squares	$\frac{1}{2}t^2$	t	1	$\|t\| < \infty$
Huber	$\frac{1}{2}t^2$	t	1	$\|t\| \leq k$
	$k\|t\| - \frac{1}{2}k^2$	$ksgn(t)$	$k/\|t\|$	$\|t\| > k$
Bi-weight	$\frac{B^2}{6}\left(1 - \left[1 - \left(\frac{t}{B}\right)^2\right]^3\right)$	$t\left[1 - \left(\frac{t}{B}\right)^2\right]^2$	$\left[1 - \left(\frac{t}{B}\right)^2\right]^2$	$\|t\| \leq B$
	$\frac{B^2}{6}$	0	0	$\|t\| > B$

With a weight matrix, Eq. (4) can be written as:

$$\mathbf{WAq} = \mathbf{0} \tag{6}$$

where

$$\mathbf{W} = \begin{pmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_N \end{pmatrix} \tag{7}$$

$$\mathbf{A} = \begin{pmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ x'_2x_2 & x'_2y_2 & x'_2 & y'_2x_2 & y'_2y_2 & y'_2 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_Nx_N & x'_Ny_N & x'_N & y'_Nx_N & y'_Ny_N & y'_N. & x_N & y_N & 1 \end{pmatrix} \tag{8}$$

the vector $\mathbf{q}$ which minimizes $\sum \rho(t)$ can be found using Singular Value Decomposition (SVD) method. The SVD of $\mathbf{WA}$ can be written as:

$$\mathbf{WA} = \mathbf{U\Lambda V}^T \tag{9}$$

where $\mathbf{\Lambda}$ is a diagonal matrix with s_j being its diagonal elements. $s_j, j = 1, 2, \ldots, 9$, is the eigenvalue of the $\mathbf{A}$ matrix, and the matrices $\mathbf{U}$ and $\mathbf{V}$ are each orthogonal in the sense that their columns are orthonormal, i.e.,

$$\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I} \tag{10}$$

Since $\mathbf{V}$ is square shaped, it is also row-orthonomal, i.e., $\mathbf{V}\mathbf{V}^T = \mathbf{I}$. N is the number of observations—the obtained corresponding points. The eigenvector of $\mathbf{V}$ which corresponds to the smallest non-zero eigenvalue in $\mathbf{\Lambda}$ is the solution of the weighted least squares. The residuals of the equations are obtained by multiplying the solution of $\mathbf{q}$ by $\mathbf{A}$. Gross errors are not necessarily accompanied by large residuals as explained in [9]. Then the residuals need to be modified by:

$$f_i(\epsilon_i) = \frac{\epsilon_i}{\sqrt{1 - h_{ii}}} \tag{11}$$

where ϵ_i is the residual for the i^{th} equation, and h_{ii} is the diagonal elements of the projection matrix $\mathbf{H}$ [5], which was also called the Hat matrix in [8]. h_{ii} is referred as the *leverage* of its corresponding data, and it is in the range of $[\frac{1}{n}, 1]$. In [7], Jones *et al* use a robust estimation technique to find the parameters of ellipse, but they do not consider the effect of leverage points. The Hat matrix of $\mathbf{A}$ is defined as:

$$\mathbf{H}_a = \mathbf{A}\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T \tag{12}$$

In analogy to Eq. (9), the SVD of the matrix $\mathbf{A}$ is:

$$\mathbf{A} = \mathbf{U}_a\mathbf{\Lambda}_a\mathbf{V}_a^T \tag{13}$$

where $\mathbf{U}_a$, $\mathbf{\Lambda}_a$ and $\mathbf{V}_a$ have the same property as $\mathbf{U}$, $\mathbf{\Lambda}$ and $\mathbf{V}$. Substituting Eq. (13) into (12), one obtains:

$$\mathbf{H}_a = \mathbf{U}_a\mathbf{U}_a^T \tag{14}$$

To consider the leverage effects, the SVD method is modified by multiplying the $\mathbf{A}$ matrix with a weight matrix as shown in Eq. (6), which is related to the residual of each equation and the leverage values. The diagonal elements of the weight matrix in Eq. (7) can be obtained as follows:

$$w_i = \sqrt{\frac{\psi(r_i^*)}{r_i^*}}\sqrt{1 - h_{ii}}, \qquad i = 0, 1, \cdots, N \tag{15}$$

where $r_i^* = \frac{r_i}{\hat{\sigma}\sqrt{1-h_{ii}}}$, h_{ii} is the diagonal elements of $\mathbf{H}_a$, and it is determined only by the values of the matched points. The object function can be modified as:

$$F = \sum_{i=0}^{N-1} \rho(r_i^*)(1 - h_{ii})\hat{\sigma} \tag{16}$$

By evaluating this function during iteration, the weighting coefficient for each measurement can be obtained when the process terminates.

4 Practical Considerations

When the number of measurements is relatively small (say, less than 30), the term $(1 - h_{ii})$ works well. In this case, if h_{ii} is large, the calculated residual by Eq. (11) is large, so the corresponding pair of points will be classified as an outlier or mismatch. But if the total number of match points is very large (say, more than 100) and if the mismatches involved are more than two, the term $(1 - h_{ii})$ will not contribute much. Because of the inequality relation $1/n < \sum h_{ii} < p$ [8], the value of every h_{ii} is very small. In this case, mismatches are difficult to identify. An alternative way to detect mismatches is to check the value of Nh_{ii}/p, where p equals to 9, the number of elements of the $\mathbf{Q}$ matrix. If this value is larger than 2, then the corresponding match can be seen as a mismatch.

With any iterative method, choosing the starting value is an important issue. From a good starting value, a procedure will converge in fewer iterations and incur less computation cost. Iterative M-estimators are particularly sensitive to the starting value when the ψ-function redescends. Given a poor starting value, an estimator with a non-monotonic ψ-function may converge to a root of equation far from the overall minimum of the objective. The strategy adopted in this algorithm is: Use a monotonic ψ-function (a Huber M-estimator) with the ordinary least squares estimate as the starting value, iterate to convergence, and then use the nonmonotonic ψ-function (a Bi-weight estimator) to iterate a few steps (perhaps only one) further.

In practice, the error scale σ must be estimated. One reason for estimating the scale is that some knowledge of it is necessary to judge the accuracy of the fitted regression model. A second reason is that, without taking scale into account, most M-estimators would not response correctly to a change in the measurement [8]. The most commonly used resistant scale of an estimator is the median absolute deviation (MAD), which is

$$\sigma = \frac{1.0}{0.6745} \operatorname*{median}_{i} \left(\left| r_i^{(0)} - \operatorname*{median}_{j} \left(r_j^{(0)} \right) \right| \right) \tag{17}$$

where $r_i^{(0)}$ is a preliminary residual of Eq. (6) and 0.6745 is the average value of the MAD for samples from the standard Gaussian distribution. Termination follows when the decrease in function F becomes small, for instance, when $F_i > \alpha F_{i-1}$, where $\alpha = 0.999$.

The steps of the algorithm involves: 1) calculating the coefficient matrix of Eq. (4) from the matched points; 2) obtaining the Hat matrix of $\mathbf{A}$ by using Eq. (12); 3) setting the initial weight matrix to unity and use the Single Value Decomposition (SVD) method to solve the weighted least squares equation; 4) based on the result of this solution, calculating the weighted matrix $\mathbf{W}$ by Huber's estimator, and evaluating the object function. If the value of the object function is less than a threshold or if the change of this value compared to the previous one is less than a threshold, we say this part of iteration has converged. Otherwise, use the the new $\mathbf{W}$ matrix to solve the least squares equation and repeat this process till the iteration converges; and 5) the last stage similar to stage 4), but calculating the $\mathbf{W}$ matrix by the Tukey's estimator.

5 Simulations and Image Tests

For the evaluation of the robust estimator, several simulations and real image tests were carried out. The effect of gross errors, and that of mismatches to the linear and the robust algorithms were evaluated in the following subsections. The points are generated randomly in a cube ($100 \times 100 \times 100$ mm). The camera is roughly 100 mm away from the bulk center of the generated points. The effective focal length of the camera is 100 mm. The image points are obtained by projecting the 3-D points inside the cube onto the image plane.

5.1 The Effects of Gross Errors

The gross errors were obtained by adding two different levels of noise to the corresponding points. Noise with 0.2 pixel variance was first added, and then 3.5, 7.0 pixels variance noise were added in the case of Figure 1 and 2 respectively. Figure 1 shows the effects of gross errors to the linear algorithm, while Figure 2 gives the effects to the robust algorithm. It can be seen from these two figures that the robust algorithm has better immunity to gross errors, and that the change of the levels of the gross errors has more effects in the linear algorithm than that in the robust algorithm.

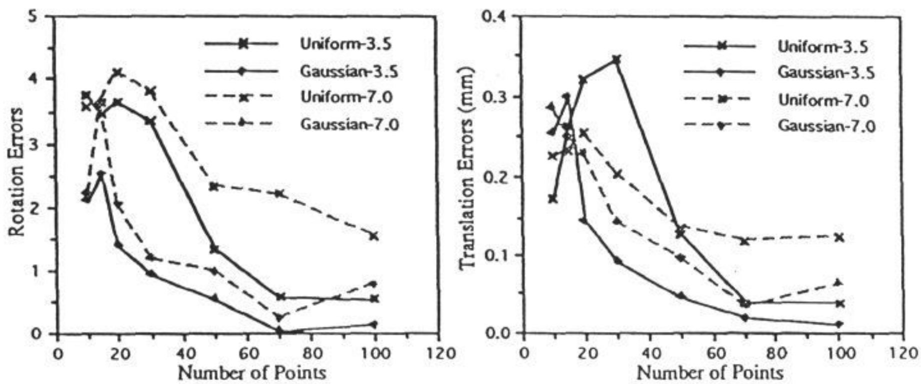

Figure 1: *The errors from different number of points for the* **linear algorithm**. *The noise level is 0.2 mm variance. Two levels of gross errors (3.5 and 7.0 pixels) are added. (***left***) The rotation error. (***right***) The translation error.*

5.2 The Effects of Mismatches

The mismatches in the whole corresponding points were obtained by randomly choosing points which are not the correct matches. The number of mismatches is 10% of the total mathched points. Figure 3 represents the effects of mismatches to the linear and the robust algorithms (left: rotation error, right: translation error). The results show that the robust algorithm gives better performance than that of the linear algorithm. Actually, the robust estimator detects mismatches and removes them automatically when calculating the Q matrix. Thereafter, the rotation matrix, the translation vector and the 3-D results can be obtained.

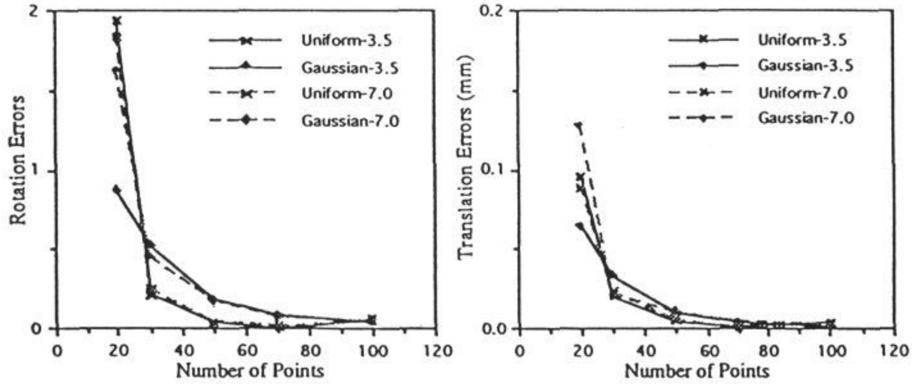

Figure 2: *The errors from different number of points for the* **robust algorithm**. *The noise level is 0.2 mm variance. Two levels of gross errors (3.5 and 7.0 pixels) are added. (***left***) The rotation error. (***right***) The translation error.*

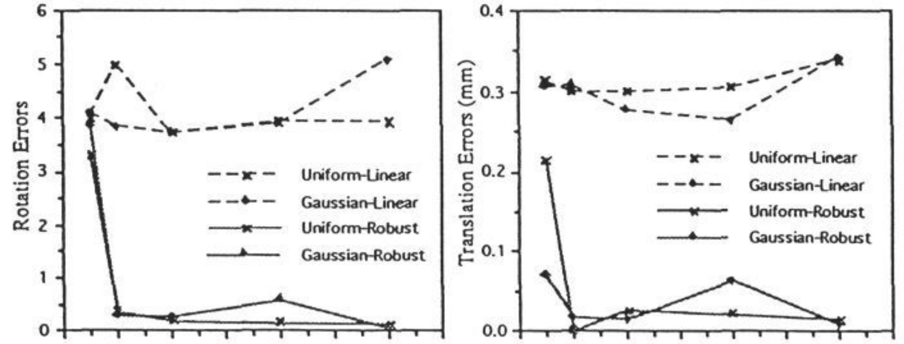

Figure 3: *The errors from different number of points for the different algorithms in the situations when mismatches occur. The noise level is 0.2 mm variance. The percentage of mismatch is 10%. (***left***) The rotation error by linear and robust algorithms. (***right***) The translation error by linear and robust algorithms.*

5.3 The Allowance of Mismatches

It can be seen from Figure 2 and Figure 3 that the robust estimator has the ability to remove gross errors and mismatch points automatically. The linear algorithm cannot deal with outliers or mismatches. The breakdown point(one can define the breakdown point of an estimator as the smallest amount of contamination that may force the value of the estimate off to arbitrary values) for the robust estimator is $1/(1+p)$ [8]. In our case, $p = 9$ (the nine unknowns for the Q matrix), then the breakdown point for the algorithm is ten per cent. That is 10% mismatched points can be allowed in the whole corresponding points without affecting the final results.

5.4 Image Tests

The algorithm described above has been tested on both synthetic and real images. The matching points are obtained by Chan's method [10], and they are shown in Figure 4(**left**). The results after the robust estimation for the mismatch removal are given in Figure 4(**right**). It can be seen that a number of edge points have been removed (e.g. the points near some of the corners, especially the corner behind the small cube). Figure 5(**left**) shows the four projections of the 3-D results obtained without mismatch removal. Figure 5(**right**) gives the 3-D results using robust estimation to achieve mismatch removal and to get more accurate 3-D structure of the objects. Comparing Figure 5(**left**) and (**right**), it can be seen that the result shown in the (**right**) are better than that shown in Figure 5(**left**). It can be seen from the top-right projection and the bottom-left projection of Figure 5(**left**) that the squares have been skewed. But the results given in Figure 5(**right**) more resemble to the real objects (squares).

Real images have also been taken to test the algorithm. The objects on the image are a cube and a headlamp model. The matching of the images was achieved by Chan's multiple view method [10]. The matched edge points before and after the robust estimation for mismatch removal are shown in Figure 6(**left**) and Figure 6(**right**) respectively. Figure 7(**left**) shows the projections of the 3-D result obtained from 2-D corresponding points before mismatch removal. Figure 7(**right**) gives the projections of 3-D result obtained by the method developed in this paper. It can be seen that the mismatches (or matches with gross errors) have been removed, and therefore the result looks clearer and better (the square in the bottom-left quarter of Figure 7(**left**) has been skewed, but it has not been skewed in Figure 7(**right**). An erroneous line in the top-right quarter of Figure 7(**left**) has been removed in Figure 7(**right**)).

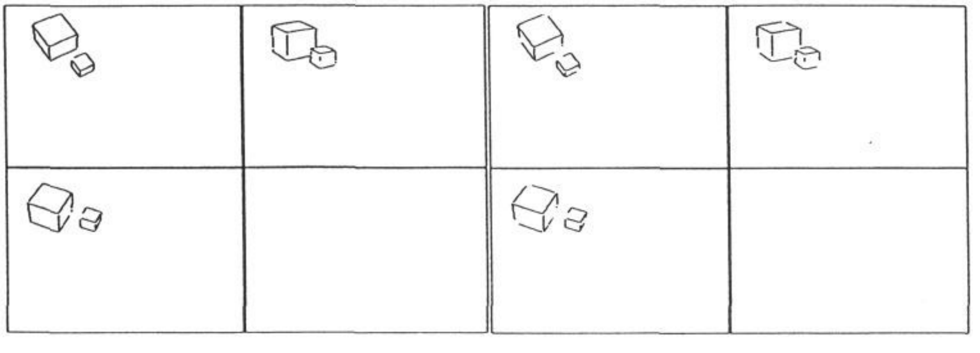

Figure 4: (**left**) *The matched edge maps of the simulated cubes;* (**right**) *The edge maps after mismatch removal from the simulated cubes.*

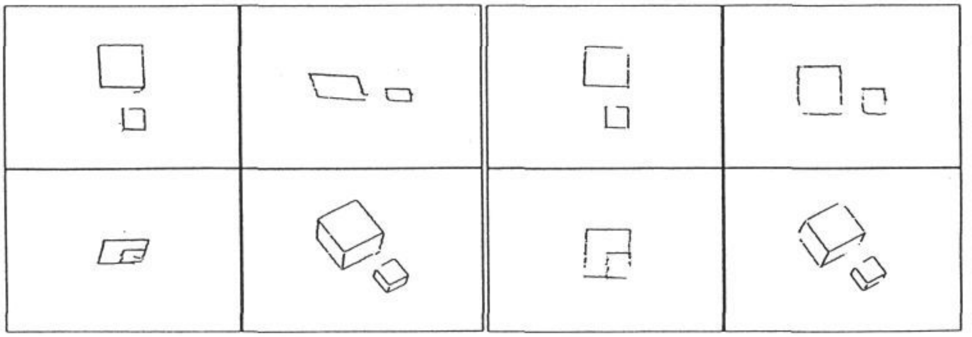

Figure 5: (left) *Projections of the results of simulated objects obtained without noise removal;* (right) *Projections of the results of simulated objects obtained using robust estimation to achieve mismatches removal.*

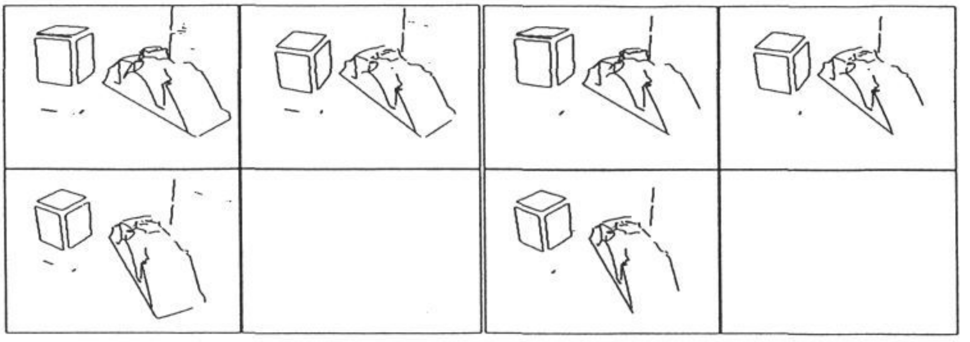

Figure 6: (left) *The edge map of the objects (a cube and a headlamp) after matching using Chan's method;* (right) *The edge map of the objects (a cube and a headlamp) after mismatch removal.*

6 Conclusions

The robust algorithm has better performance than the linear algorithm when gross errors and mismatches occur, because the robust algorithm can detect outliers and mismatches and remove them automatically. The existence of a small number of outliers or mismatches will not affect the final results. Different robust estimators can be used for the purpose of parameters estimation. In our case the Huber and Tukey's estimators are used which allow ten per cent mismatches. The median absolute estimator can also be used which can allow as high as fifty per cent outliers in the corresponding points. But one of the disadvantages is that this takes much more time.

Robust estimation offers a theoretical framework for assessing outlier rejection schemes, and more importantly, provides an approach capable of great accuracy to parameter estimation in contaminated data distributions. These methods will have wider applications in computer vision and image processing.

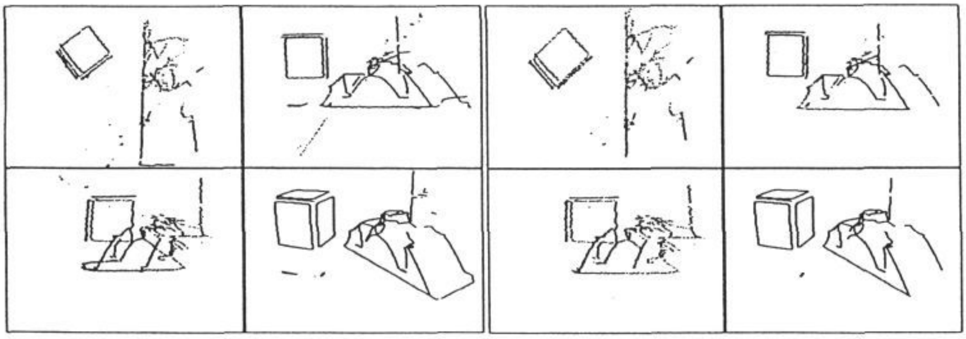

Figure 7: (**left**) *Results obtained from data with mismatches;* (**right**) *Results of real objects obtained using robust estimation to achieve mismatches removal.*

References

[1] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[2] R. Y. Tsai and T. S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6:13–26, January 1984.

[3] E. H. Thompson. A rational algebraic formulation of the problem of relative orientation. *Photogrammetric Record*, 3:152–159, 1959.

[4] T. S. Huang and Y. S. Shim. Linear algorithm for motion estimation: How to handle degenerate cases. In *The Forth International Conference on Pattern Recognition*, pages 439–447, 1988.

[5] C. N. Lee, R. M. Haralick, and X. Zhuang. Recovering 3-D motion parameters from image sequences with gross errors. In *Proceedings of Workshop on Visual Motion*, pages 46–53, Irvine, CA, USA, 20–22 March 1989.

[6] P. J. Besl, J. B. Birch, and L. T. Watson. Robust window operators. In *Second International Conference on Computer Vision*, pages 591–600, 1988.

[7] G. A. Jones, J. Princen, J. Illingworth, and J. Kittler. Robust estimation of shape parameters. In *First British Machine Vision Conference*, pages 43–48, Oxford, UK, 1990.

[8] G. Li. Robust regression. In D. C. Hoaglin, F. Mosteller, and J. W. Tukey, editors, *Exploring Data Tables, Trends, and Shapes*, chapter 8, pages 281–343. John Wiley & Sons, Inc., N.Y., 1985.

[9] P. J. Huber. *Robust Statistics*. John Wiley & Sons, Inc., New York, 1981.

[10] K. L. Chan. *Optimization of Multiple View Stereo Vision*. PhD thesis, Imperial College, University of London, Exhibition Road, London SW7 2BX, UK, 1991.