

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/224557228>

An adaptive-scale robust estimator for motion estimation

Conference Paper in Proceedings - IEEE International Conference on Robotics and Automation · June 2009

DOI: 10.1109/ROBOT.2009.5152445 · Source: IEEE Xplore

CITATION

1

READS

35

6 authors, including:



Ngo Thanh Trung

Osaka University

24 PUBLICATIONS 245 CITATIONS

[SEE PROFILE](#)



Hajime Nagahara

Kyushu University

125 PUBLICATIONS 913 CITATIONS

[SEE PROFILE](#)



Ryusuke Sagawa

National Institute of Advanced Industrial Science and Technology

130 PUBLICATIONS 1,459 CITATIONS

[SEE PROFILE](#)



Yasushi Yagi

Osaka University

366 PUBLICATIONS 4,559 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Human behavior Analysis [View project](#)



Underwater 3D scanning [View project](#)

An Adaptive-Scale Robust Estimator for Motion Estimation

Trung Ngo Thanh^a, Hajime Nagahara^a, Ryusuke Sagawa^a,
Yasuhiro Mukaigawa^a, Masahiko Yachida^b, Yasushi Yagi^a

^a Osaka University, ^b Osaka Institute of Technology

Abstract—Although RANSAC is the most widely used robust estimator in computer vision, it has certain limitations making it ineffective in some situations, such as the motion estimation problem, in which uncertainty on the image features changes according to the capturing conditions. The greatest problem is that the threshold used by RANSAC to detect inliers cannot be changed adaptively; instead it is fixed by the user. An adaptive scale algorithm must therefore be applied in such cases. In this paper, we propose a new adaptive scale robust estimator that adaptively finds the best solution with the best scale to fit the inliers, without the need for predefined information. Our new adaptive scale estimator matches the residual probability density from an estimate and the standard Gaussian probability density function to find the best inlier scale. Our algorithm is evaluated in several motion estimation experiments under varying conditions and the results are compared with several of the latest adaptive-scale robust estimators.

I. INTRODUCTION

Most motion estimation algorithms in computer vision rely on the detection of image feature points such as the KLT feature [1][2], Harris feature [3], or SIFT feature [4], and so on. There is, however, always a large degree of uncertainty in vision such as noise in the image, occlusion and moving objects, and therefore a number of outlying features exist that do not follow the motion function. The most popular method to ignore those features that do not meet the constraints is to apply a robust estimator to eliminate such features and retain the reliable ones. RANSAC [5] is widely used in this regard. RANSAC assumes that the scale of inliers is known and uses this to distinguish the inliers from outliers. RANSAC and its improvements [6][7] have successfully been applied in various motion estimation applications such as camera egomotion [8][9], structure from motion [10] or simultaneous localization and map building [11], and motion segmentation [12].

Besides large uncertainty, smaller uncertainty also affects the inlying features, in that the features still follow the motion constraint, but with some error. In the event of fixed conditions, such as the same camera, same image resolution and similar motion speed, the user-defined threshold supplied to RANSAC is sufficient. However, in other applications where this uncertainty varies or where the correct threshold is not known. For example, the video is sometimes blurred due to the fast motion of the camera. An adaptive-scale estimator needs to be applied in such situations.

In this paper, we present a new robust estimator that can work in high outlier-rate data and estimates the correct inlier threshold. Our method relies on a new inlier scale estimator and a kernel density estimation-based objective function. We

assume that the residual distribution is Gaussian as is the case in most previous works. The algorithm for the new inlier scale estimator, which is our main contribution in this paper, matches the Gaussian distribution and the residual distribution from a putative solution. One of the advantages of this inlier scale estimator is that inliers and outliers are well-detected; the other advantage is that it does not depend heavily on the bandwidth of the residual density estimation.

II. RELATED WORKS

Several robust estimators that can adjust the inlier scale adaptively and do not need predefined information have been proposed in computer vision. The requirements for such robust estimators are: robustness with respect to high outlier-rate (or high breakdown point [13]) and a sufficient number of inliers. The LMS (least median of squares) [13] is the most widely known estimator that can be used in these situations, but it can only be applied when the outlier-rate is less than 50%. Some extensions of LMS like MUSE (minimum unbiased scale estimate) [14] or ALKS (adaptive least kth order squares) [15] can be applied under high outlier-rates, however these have a problem with extreme cases, such as perpendicular planes. Another extension of LMS is MINPRAN (minimize probability of randomness) [16], which requires an assumption of the outlier distribution. This assumption seems to be strict since outlier distribution is assumed with difficulty. The pbM (projection-based M-estimator) [17][18][19] is an extension of the M-Estimator that uses projection pursuit and the kernel density estimation (KDE), and can provide a breakdown point much greater than 50%. However, it only works for linear residual functions, such as linear regression, and uses a bandwidth (some proportion of MAD (median absolute deviations) scale estimate) which is robust to only 50% of the outlier-rate, and thus the robustness is reduced. Another robust estimator that uses KDE is the ASSC (adaptive scale sample consensus) [20]. ASSC assumes that the inliers are located within some special structure of the density distribution; it practically detects a first peak from zero and a valley next to the peak to locate the inliers. ASSC can provide a very high breakdown point, around 80%, when applying the proper bandwidth. However, the bandwidth is undersmoothed when there are only a few or no outliers and becomes oversmoothed with a large number of outliers. ASSC has subsequently been improved as the ASKC (adaptive scale kernel consensus) [21]. ASKC improves the objective function of ASSC using KDE and improves the robustness in the case of a high

outlier-rate. The bandwidth of KDE in ASKC is computed using a scale that contains approximately 10% of the smallest residuals. However, this undersmoothed bandwidth means that the ASKC can estimate very few inliers in the case of data with a low outlier-rate.

III. OVERVIEW OF PROPOSED ESTIMATOR

Similar to previous diagnostic robust estimators, the proposed estimator consists of two important components: an inlier scale estimator and an objective function. We also use a statistical random sampling method for the search procedure as in RANSAC. The inlier scale estimator uses matching by correlation between the standard Gaussian distribution (SGD) and the residual distribution of a putative estimate; the scale that gives the highest correlation will become the inlier scale estimate. The objective function is based on the kernel density estimation to evaluate the density of estimated inliers; the solution with the largest value is then the output of the estimator. The details are described in the next section.

IV. ADAPTIVE-SCALE ROBUST ESTIMATOR

A. Problem Preliminaries

Assume the estimation of a model with the constraint:

$$g(\theta, \mathbf{X}_i^t) = 0, \quad (1)$$

where θ is the ideal parameter vector that describes the outcome of inliers, and \mathbf{X}_i^t is the ideal data point without noise. Our estimation problem is then described as:

- Input: There are N observed data points $\mathbf{X}_i, i = 1..N$. These N observed data points include both inliers and outliers.
- Output: Parameter θ that describes the data.

In a real problem, each inlier \mathbf{X}_i^t is affected by some unknown noise \mathbf{n}_i :

$$\mathbf{X}_i = \mathbf{X}_i^t + \mathbf{n}_i. \quad (2)$$

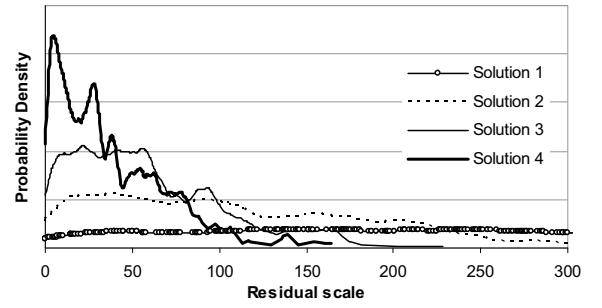
Therefore, the ideal parameters θ cannot be recovered, and some approximation of θ will be estimated. A robust estimator based on random sampling like RANSAC solves the problem by trying many random solutions $\hat{\theta}$, with the best solution $\hat{\theta}^*$ being the approximation of θ . To evaluate a solution $\hat{\theta}$ good or bad, the estimator can only rely on the statistics of the error for each data point; this error is called the residual [13]. For each model estimation problem, there are numerous ways to define the residual function, including using the original constraint function (1). Generally, however, the residual is defined as:

$$r_{i,\hat{\theta}} = f(\hat{\theta}, \mathbf{X}_i). \quad (3)$$

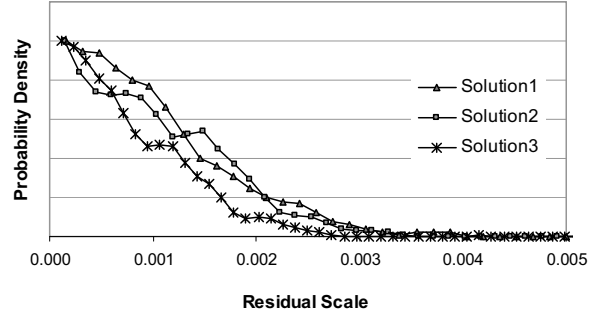
A good definition of the residual is that proposed by Q.T. Luong et al. [22]:

$$r_{i,\hat{\theta}} = \frac{g(\hat{\theta}, \mathbf{X}_i)}{\|\nabla g(\hat{\theta}, \mathbf{X}_i)\|}, \quad (4)$$

where $\nabla g(\hat{\theta}, \mathbf{X}_i)$ is the gradient of g with respect to variable \mathbf{X}_i . In our algorithm we use the absolute residual.



a) Distribution for Line Fitting



b) Distribution for Fundamental matrix estimation

Fig. 1. Distribution of absolute residuals in several good solutions (in increasing order of accuracy) for (a) line fitting and (b) the fundamental matrix estimation problem, assuming that a zero-mean Gaussian noise model influences the inliers. These distribution curves correlate strongly with the standard Gaussian distribution curve.

In an ideal case, there is no noise on the inliers, and thus the residual from an inlier is zero. However, in a real problem, an inlier is always affected by some noise and the estimated solution $\hat{\theta}$ is not always correct. The inlier residual is therefore, not zero and is computed by (3). The standard deviation of these inlier residuals is called the “inlier scale”, and is denoted by $\sigma_{\hat{\theta}}$. The problem is that $\sigma_{\hat{\theta}}$ is not known, and therefore, our proposed inlier scale estimator tries to estimate it. This estimate is denoted by $\sigma_{\hat{\theta}}^*$. Once the inlier scale has been found, the inlier threshold $t_{\hat{\theta}}$ can be decided to distinguish inliers from outliers.

B. Inlier Scale Estimator

Statistically, for an accurate solution $\hat{\theta}$, the distribution of residuals from inliers is densely distributed around zero and sparsely distributed for outliers. The more accurate the solution, the denser the distribution of inlier residuals is. This is a well-known phenomenon for any estimation problem. We demonstrate this in Fig.1.

We can see from Fig.1 that the distribution of residuals from inliers is strongly related to the standard Gaussian distribution. This is also widely assumed by previous similar works. Therefore, we propose an algorithm for density matching to find the best inlier scale.

1) *Matching of Residual Distribution and Ideal Distribution:* The inlier scale is estimated by finding the best correlation between a segment of the residual distribution and the SGD. The segment of the residual distribution used

for matching starts from zero. Then, the scale of the first structure is detected regardless of the outlier structures. The similarity between the SGD curve $P^s(\alpha)$ and a normalized residual distribution curve $P_{\hat{\theta}}^s(\frac{r_i}{\sigma})$, given an assumed noise standard deviation σ of inliers, is defined from the correlation coefficient. This similarity is a function of σ :

$$s_{\hat{\theta}}(\sigma) = \text{Corr}_{0 \leq x_i \leq \kappa \sigma} (P_{\hat{\theta}}^s(\frac{x_i}{\sigma}), P^s(\frac{x_i}{\sigma})), \quad (5)$$

where x_i is the scale variable and κ indicates the part of the SGD used in the matching. For example, κ is defined so that the standard cumulative distribution function Φ of the SGD has the value $\Phi(\kappa) = 0.997$, which means that the SGD segment containing 99.7% of the samples is used for fitting; in this case $\kappa = 3$. By this definition, similarity is limited to $-1 \leq s_{\hat{\theta}}(\sigma) \leq +1$. Then, the best scale of noise on inliers $\sigma_{\hat{\theta}}^*$ is estimated by searching the scale that gives the highest similarity, which is summarized as

$$\sigma_{\hat{\theta}}^* = \underset{\sigma}{\operatorname{argmax}} \{s_{\hat{\theta}}(\sigma)\}. \quad (6)$$

Inliers are then distinguished using the inlier threshold $t_{\hat{\theta}} = \kappa \sigma_{\hat{\theta}}^*$. In our algorithm, to compute the probability of the residual from an estimate, we apply the well-known histogram method, although the KDE can also be used. A histogram is simple and gives us low computational cost. Then, the variable x_i in (5) is the bin variable. Searching for the best inlier threshold $t_{\hat{\theta}}$ is graphically depicted in Fig.2 and Fig.3.

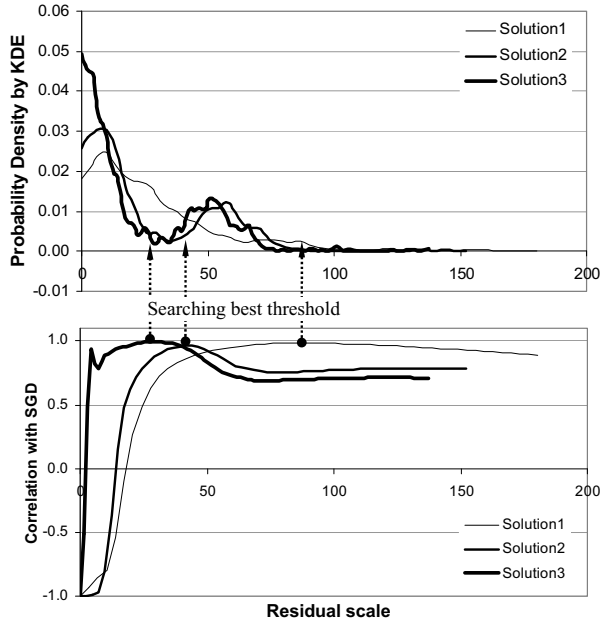


Fig. 2. Line fitting: Several residual distributions and their fitness with respect to the SGD. The inlier threshold $t_{\hat{\theta}} = \kappa \sigma_{\hat{\theta}}^*$ is obtained by finding the highest correlation.

Since the correlation is not very stable at the initial scales since few bins are used for the correlation, as shown in Fig.2 and Fig.3, we can easily avoid these cases by checking the actual probability density at the best threshold $t_{\hat{\theta}}$. At the

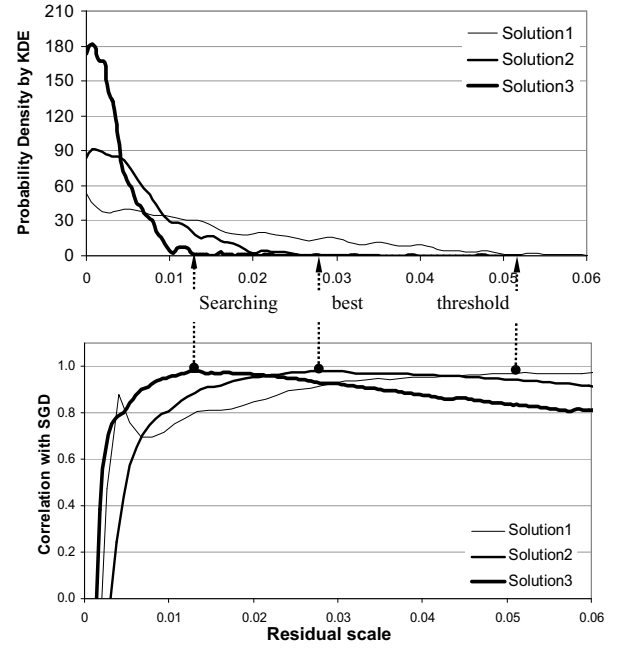


Fig. 3. Fundamental matrix estimation: Several residual distributions and their fitness with respect to the SGD. The inlier threshold $t_{\hat{\theta}} = \kappa \sigma_{\hat{\theta}}^*$ is obtained by finding the highest correlation.

scale of the best threshold $t_{\hat{\theta}}$, the probability density must be low:

$$P_{\hat{\theta}}(t_{\hat{\theta}}) \leq 0.2 \max_{r < t_{\hat{\theta}}} P_{\hat{\theta}}(r). \quad (7)$$

2) *Bin-width Selection*: Bin-width is size of bin in the residual histogram. In this section, we decide the bin-width that is used in our algorithm. Bin-width is usually a difficult problem for those methods that rely on the probability density of residuals. A widely used bin-width [23] for robust estimators is:

$$\hat{h} = \left(\frac{243 \int_{-1}^1 K(\zeta)^2 d\zeta}{35N (\int_{-1}^1 \zeta^2 K(\zeta) d\zeta)^2} \right)^{\frac{1}{5}} \hat{\sigma}, \quad (8)$$

where K is some kernel, such as the popular Gaussian kernel, Epanechnikov kernel, etc. and $\hat{\sigma}$ is some scale estimate, such as the standard deviation of residuals, median scale estimate [13] or MAD estimate [13], and N is the number of data points. The standard deviation scale estimate is very sensitive to outliers and is rarely used. Median and MAD scale estimates are robust up to 50% outliers and are frequently used in robust estimators. Where the outlier-rate is greater than 50%, these scale estimates are badly affected by the large residuals of outliers.

In the proposed algorithm, we fit the residual histogram to the ideal distribution, the Gaussian distribution, and the histogram does not need to be as smooth as in previous works based on the KDE. The best overall correlation will result in the scale estimate. Moreover, the objective of a recent robust estimator is to deal with a high outlier-rate, say 80%. Consequently, in the proposed method, the histogram bin-width is estimated using a scale estimate $\hat{\sigma}$ that is the

smallest window containing 15% of the smallest residuals. Where the outlier-rate is less than 80%, some of the leftmost bins will contain inliers, a situation that can be detected by our algorithm.

Having obtained the bin-width, a histogram of the estimate can be built. Since the bin-width is small to avoid over-smoothing in the case of a high outlier-rate, the number of bins may be large, and therefore in practice, we have to prune all unnecessary bins. Pruning can be done by removing the low density bins from the rightmost bin.

C. Objective Function

Inspired by the use of the KDE in the pbM-Estimator [18][19], ASSC [20] and ASKC [21], we also apply it in our adaptive objective function:

$$F(\hat{\theta}) = \frac{1}{N\kappa\hat{\sigma}_{\hat{\theta}}^*} \sum_{i=1}^N K\left(\frac{r_{i,\hat{\theta}}}{\kappa\hat{\sigma}_{\hat{\theta}}^*}\right), \quad (9)$$

where $\hat{\sigma}_{\hat{\theta}}^*$ is adaptively estimated by the proposed inlier scale estimator as shown in Section IV-B.1 and κ has been defined in Section IV-B.1; K is a kernel such as Gaussian or Epanechnikov kernel. The KDE objective function evaluates how densely the residuals are distributed at zero using a kernel's window. In our case, the window of kernel K is $\kappa\hat{\sigma}_{\hat{\theta}}^*$, which covers all the estimated inliers, therefore the objective function gives the density measured at zero only for inliers. Similar to the M-estimators, a large residual makes a small contribution, whereas a small residual makes a large contribution to the overall score. However, this objective function is different to that in the conventional M-Estimators in two aspects. First, the scale estimate $\hat{\sigma}_{\hat{\theta}}^*$ is estimated for only inlier residuals, and is adaptively estimated. Second, the sum of weights on the residuals is scaled by $\frac{1}{\hat{\sigma}_{\hat{\theta}}^*}$, which intensifies the score when the estimated inlier scale is small and reduces the score when the estimated inlier scale is large. In summary, the KDE objective function declares a solution to be better under the following conditions:

- (a) Larger number of estimated inliers,
- (b) Smaller scale of inliers,
- (c) Smaller residuals of inliers.

D. Estimation Algorithm Summary

In this section, we summarize the proposed algorithm.

- (a) Create a random sample then estimate the solution parameters $\hat{\theta}$,
- (b) Estimate all the residuals of the data points given the model parameters $\hat{\theta}$,
- (c) Estimate the bin-width as described in Section IV-B.2, then compute the residual histogram $P_{\hat{\theta}}$. The histogram is then pruned as described in Section IV-B.2,
- (d) Estimate the inlier scale as summarized by (6) and (7),
- (e) Estimate the score using the objective function (9),
- (f) Update the best solution,
- (g) Repeat from (a) if not terminated.

A flowchart of the algorithm is given in Fig.4.

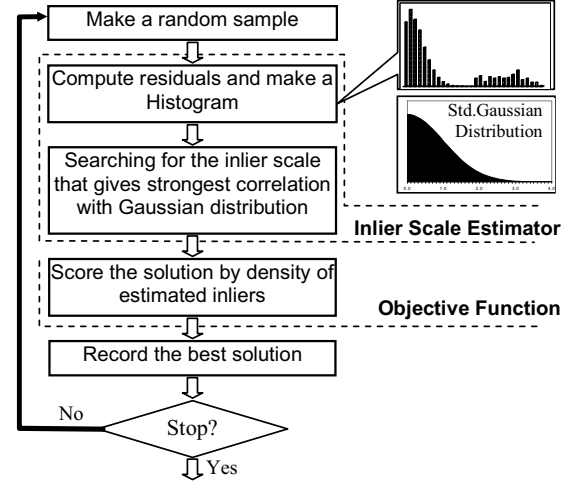


Fig. 4. Flowchart of proposed algorithm.

V. EXPERIMENTS

In this section, we describe several experiments of various popular motion estimation problems using omnidirectional vision, namely rotation estimation and fundamental matrix estimation [22]. Epanechnikov kernel was used for the objective function in all experiments. We compare our results with those of LMedS (a well-known random sampling-based implementation of LMS), ASSC, and ASKC. For ASSC, we set $c = 0.5$ as the constant for the KDE bandwidth computation. All algorithms were supplied with the same random sampling and the stop criterion was a fixed number of iterations. The results are shown as the average of 200 executions. No optimization of the estimation was applied for any algorithm. The camera was mounted on a rotation stage and rotated by a controller, and thus the rotation of the camera was known. We used KLT feature detection and tracking [2] implemented in OpenCV [24]. Examples of image sequences used in the experiments and 200 KLT features detected on each image are shown in Fig.5.

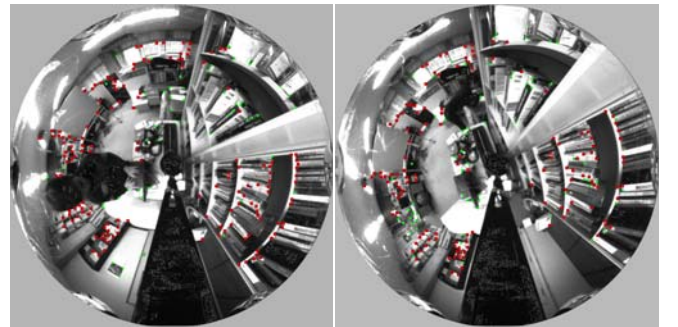


Fig. 5. A pair of matching images from the proposed estimator for rotation estimation: red and large features are matched features on both images, while green and small features are unmatched.

A. Rotation Estimation

We applied our robust estimator to estimate the rotation of the omnidirectional camera. Matching features between

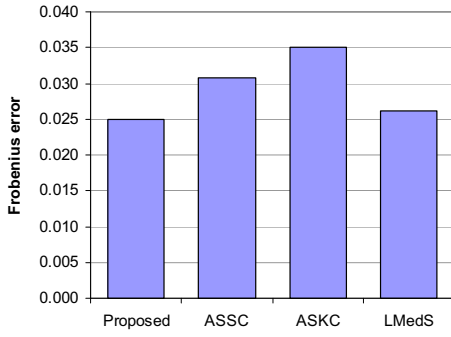


Fig. 6. Rotation estimation error (Frobenius error)

consecutive images is performed to find the feature correspondence pairs P_i, P'_i that follow only rotation R :

$$P'_i = RP_i. \quad (10)$$

Near features that do not follow the pure rotation are then eliminated in the matching. Image feature coordinates are mapped onto the unit sphere, while feature tracking for correspondence between consecutive images is not used. Feature correspondence and rotation are simultaneously estimated. The residual is defined as the angle between two vectors of feature correspondence after applying the estimated rotation:

$$r_{i,\hat{R}} = \text{Angle}(P_{m,i}, P'_{m,i}, \hat{R}), \quad (11)$$

where $(P_{m,i}, P'_{m,i})$ is a correspondence pair of matched features mapped onto the unit sphere and \hat{R} is an estimated rotation matrix. The estimation error for one pair of images is defined as the Frobenius norm between the ground-truth rotation matrix that is given by the controller. In this experiment, a sequence of about 50 frames was captured with the ground-truth rotation angle for each frame, and then 200 features were detected on each frame. An example of a matching pair from the proposed algorithm is shown in Fig.5. Average results from 200 executions, with 1500 iterations of random sampling for each execution, are shown in Fig.6. These results show that our proposed algorithm gives the best results of all compared estimators.

B. Fundamental Matrix Estimation

In this experiment, we applied the proposed algorithm to fundamental matrix estimation in simulation and real video sequences. The residual is defined as follows [22]:

$$r_{i,\hat{F}} = \frac{|x_i'^T \hat{F} x_i|}{\sqrt{\|\hat{F} x_i\|^2 + \|\hat{F}^T x_i'\|^2}}, \quad (12)$$

where \hat{F} is an estimated fundamental matrix and (x_i, x'_i) a pair of feature correspondences between consecutive images. Fundamental matrix was estimated using seven point algorithm [25]. In this section, we first estimate fundamental matrix in simulation and then in real video sequence. For these experiments, the error of the estimation between one pair of views is computed as the standard deviation of the

estimated residual of the best solution from each estimator. The better the solution, the smaller are the estimated residuals. In simulation, we know the ground-truth inliers. For real experiments, the error is computed for M fixed smallest residuals. M is minimum number of correct correspondence from feature tracker for all video sequences.

1) *Fundamental Matrix Estimation in Simulation:* We simulated points on a unit sphere, 500 points were distributed randomly on a unit sphere. Some motion of the view point was made resulted in the motion of these points on the sphere, then we had 500 pairs of point correspondence. Some pairs of these pairs were then replaced by outlying pairs with the random point coordinates so that total number of pairs was constantly 500. Coordinates (x, y, z) of each inlier point on the unit sphere before and after the motion were contaminated by Gaussian noise with zero mean and noise standard deviation σ_G . Experiments of the estimation errors under various outlier-rates were carried out and the average results of 100 trials are shown in Fig.7, with $\sigma_G = 0.005$, 10000 random samples for each estimator for one trial. The results prove that our proposed algorithm gives the overall best robustness under high outlier-rates. For low outlier-rates less than 50%, LMedS gave the best performance. However, for high outlier-rates, our proposed estimator gave the best results.

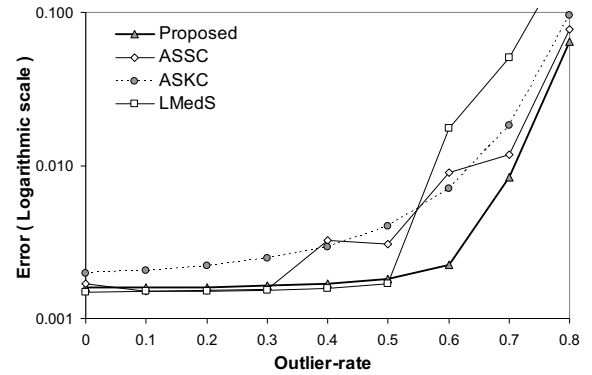


Fig. 7. Fundamental matrix estimation error in simulation.

2) *Fundamental Matrix Estimation for Real Video Sequence:* Features were tracked using the KLT tracker and then mapped onto the unit sphere; 200 features per frame were used. Several video sequences of the same scene were captured with different control speeds as shown in Fig.5. The conditions for the experiment were low or high speed of the controller and with or without motion blur. Thus in total four video sequences were captured. Both motion blur and high speed contaminated the tracking of features. For low and high speed without motion blur, the rotation velocity between consecutive images was 15 deg/frame and 20 deg/frame, respectively, while for low and high speed with motion blur, the rotation velocity was about 27 deg/frame and 42 deg/frame, respectively. Translation of the camera was done simultaneously with the rotation controlled by the same controller. High rotation velocities resulted in high outlier-rates of the input data.

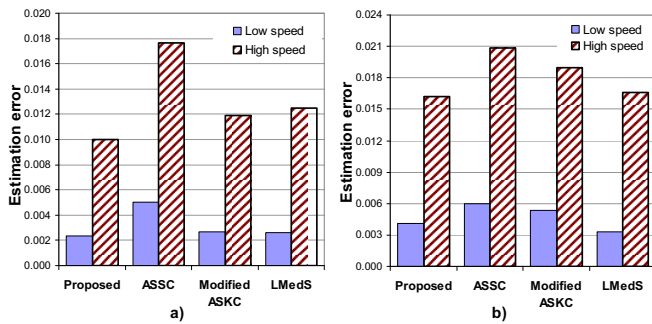


Fig. 8. Fundamental matrix estimation error: (a) two sequences with different control speeds and without motion blur and (b) two sequences with different control speeds and with motion blur.

Since each estimator estimates a different number of inliers, we normalize this error by applying a fixed number of smallest estimated residuals from the estimated solution. In this experiment we used $M=70$ as the fixed number since at least 70 (out of 200) features were tracked correctly. In this experiment, the KDE bandwidth used in ASKC was found to be too undersmoothed, and therefore the ASKC did not work correctly. We modified the bandwidth of ASKC by scaling it up 5 times; this is denoted as the Modified ASKC in the graphs. The experimental results under various conditions are shown in Fig.8.a and Fig.8.b. These results also confirm that our proposed method gives the best results of all the robust estimators. The Modified ASKC worked slightly better than its ancestor, the ASSC.

With respect to computational cost, LMedS is the fastest of the estimators since it is very simple. Second fastest is the ASKC, since it applies two objective functions for scoring, the first of which filters out the earlier bad solutions. Our proposed estimator and the ASSC have a similar computational cost. In practice, one can obviously improve the computational cost by applying multiple objective functions, as is done in several previous works.

VI. CONCLUSIONS AND FUTURE WORKS

In this paper we have proposed a new highly robust estimator for the estimation problem in computer vision that deals with data with a high outlier-rate. Our algorithm does not need any prior information about the inlier scale, which is adaptively estimated. The correlation between the residual distribution and the standard Gaussian distribution is key to enabling us to detect the inlier scale. The experiments with several motion estimation problems have positively validated our proposed algorithm. The residual distribution of inliers for motion estimation is not strictly the Gaussian distribution, however, the correlation between this distribution and the Gaussian distribution is much stronger than the correlation between the residual distribution of outliers and the Gaussian distribution.

The proposed robust estimator has shown some initial promising results as illustrated in this paper. In future work, we aim to analyze our method for other estimation problems and improve the method.

REFERENCES

- [1] B.D. Lucas and T. Kanade, An iterative image registration technique with an application to stereo vision, IJCAI81, pp. 674–679, 1981.
- [2] J. Shi and C. Tomasi, Good Features to Track, Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 593–600, 1994.
- [3] C. Harris and M.J. Stephens, A combined corner and edge detector, Proc. of the 4th Alvey Vision Conference, pp. 147–151, 1988.
- [4] David G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision, 60,2, pp. 91–110, 2004.
- [5] M. A. Fischler and R. C. Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, Comm. of the ACM 24, pp. 381–395, 1981.
- [6] D.Nister, Preemptive RANSAC for live structure and motion estimation, Ninth IEEE International Conference on Computer Vision, pp. 199, 2003.
- [7] J. Matas and O. Chum, Randomized RANSAC with Sequential Probability Ratio Test, Proc. International Conference on Computer Vision, vol. 2, pp.1727–1732, 2005.
- [8] R. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, 2000.
- [9] T.T. Ngo, H. Nagahara, R.Sagawa, Y.Mukaigawa, M.Yachida,Y.Yagi, Robust and Real-Time Egomotion Estimation Using a Compound Omnidirectional Sensor, Proc. IEEE International Conference on Robotics and Automation, pp. 492–497, 2008.
- [10] P.H.S. Torr and A. Zisserman, Feature Based Methods for Structure and Motion Estimation, Lecture Notes in Computer Science, Springer Berlin/Heidelberg, 2000.
- [11] E.Eade, and T.W.Drummond, Edge landmarks in monocular SLAM, The 17th British Machine Vision Conference, 2006.
- [12] P.H.S. Torr, D.W. Murray, Outlier detection and motion segmentation, Proc. Sensor Fusion VI, vol. SPIE 2059, pp. 432–443, 1993.
- [13] P.J. Rousseeuw and A. Leroy. Robust Regression and Outlier Detection. John Wiley & Sons, New York. 1987.
- [14] J.V. Miller and C.V. Stewart, MUSE: Robust surface fitting using unbiased scale estimates, in Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, pp. 300–306, 1996.
- [15] K.M. Lee, P. Meer, and R.H. Park, Robust adaptive segmentation of range images, IEEE Trans. Pattern Anal. Machine Intelligence, 20, pp. 200–205, 1998.
- [16] C. V. Stewart, MINPRAN: A new robust estimator for computer vision, IEEE Trans. Pattern Anal. Machine Intelligence, 17, pp. 925–938, 1995.
- [17] H. Chen and P. Meer, Robust regression with projection based M-estimators, Proc. 9th Intl. Conf. on Computer Vision, pp. 878–885, 2003.
- [18] S.Rozenfeld, I.Shimshoni, The Modified pbM-Estimator Method and a Runtime Analysis Technique for the RANSAC Family, Proc. CVPR pp. 1113–1120, 1995.
- [19] Raghav Subbarao, Peter Meer, Beyond RANSAC: User Independent Robust Regression, p. 101, 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW’06), 2006.
- [20] H. Wang and D. Suter, Robust Adaptive-Scale Parametric Model Estimation for Computer Vision. PAMI.26(11), pp. 1459–1474, 2004.
- [21] H. Wang, D. Mirota, M. Ishii, G.D. Hager, Robust Motion Estimation and Structure Recovery from Endoscopic Image Sequences With an Adaptive Scale Kernel Consensus Estimator, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2008.
- [22] Q.T. Luong and O.D.Faugeras, The fundamental matrix: Theory, algorithms, and stability analysis, Intl. Journal of Computer Vision, Vol.17, No.1, pp. 43–75, 1996.
- [23] M.P. Wand and M. Jones. Kernel Smoothing. Chapman & Hall. 1995.
- [24] Open Source Computer Vision Library, <http://www.intel.com/technology/computing/opencv/index.htm>, Intel Corporation.
- [25] R.I. Hartley, Projective reconstruction and invariants from multiple images, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 16 (10), pp.1036–1041, Oct 1994.