IPSJ Transactions on Computer Vision and Applications Vol. 1 260–276 (Nov. 2009)

Regular Paper

Highly Robust Estimator Using a Case-dependent Residual Distribution Model

Ngo Trung Thanh,[†1] Hajime Nagahara,[†1]
Ryusuke Sagawa,[†1] Yasuhiro Mukaigawa,[†1]
Masahiko Yachida[†2] and Yasushi Yagi[†1]

The latest robust estimators usually take advantage of density estimation, such as kernel density estimation, to improve the robustness of inlier detection. However, the challenging problem for these systems is choosing the suitable smoothing parameter, which can result in the population of inliers being over- or under-estimated, and this, in turn, reduces the robustness of the estimation. To solve this problem, we propose a robust estimator that estimates an accurate inlier scale. The proposed method first carries out an analysis to figure out the residual distribution model using the obvious case-dependent constraint, the residual function. Then the proposed inlier scale estimator performs a global search for the scale producing the residual distribution that best fits the residual distribution model. Knowledge about the residual distribution model provides a major advantage that allows us to estimate the inlier scale correctly, thereby improving the estimation robustness. Experiments with various simulations and real data are carried out to validate our algorithm, which shows certain benefits compared with several of the latest robust estimators.

1. Introduction

Robust parameter estimation is fundamental research in the fields of statistics and computer vision. It can be applied in many estimation problems, such as extracting geometric models in intensity images and range images, estimating motion between consecutive image frames in a video sequence, matching images to find their similarity, and so on. In these problems, the data contains explanatory data, which also includes leverage elements, and a large number of outliers. The explanatory data points are usually affected by noise with an unknown bound. For example, noise on images depends on the specific sensor, the sensor setting, and the resolution or motion blur on the captured images. The data may also contain several structures, such as various lines or planes that appear in pictures of buildings or range images, and the estimator has to extract inliers correctly to prevent multiple structures from merging into one or one structure from being divided into several smaller structures. Therefore, the common requirements for a modern robust estimator in computer vision are: robustness to various high outlier rates (high breakdown point [1)]), good detection of inliers (or good inlier bound), and the ability to work with data containing multiple structures.

Although RANSAC [2)] is an attractive robust estimator because it is simple and can tolerate more than 50% outliers, it requires the inlier noise scale from the user. Several robust estimators also based on random sample consensus have been proposed to avoid the need for user-defined parameters. These robust estimators can be classified into two groups according to whether or not they make use of residual density estimation. The first group consists of estimators [3)–5)] that search the sorted residuals for the boundary between inliers and outliers without using density estimation. On the other hand, the estimators [6)–9)] in the second group detect the inliers using the residual density estimation. A problem for the first group is that the estimators are sensitive to small pseudo-structures in the data and are less robust in real applications. The methods used in the second group apply a smoothing parameter to estimate the residual density, and they are therefore not so sensitive to small structures, but instead they have to deal with the well-known problem of density smoothness. Since the smoothing parameter for the density estimation is computed before the inlier scale is known, the density of the residual is usually over- or undersmoothed, which results in a corresponding over- or under-estimate of the proportion of inliers. On the contrary, in the experiments with the robust estimator we understand that the robustness of the estimator depends on the size of the proportion of inliers, that can be detected by the inlier scale estimator. If the inlier scale estimator detects only a few inliers or too many inliers, the robustness of the estimator is reduced, as is shown in the simulation in **Fig. 1**.

In this paper, we present a new robust estimator, an improvement for RANSAC, that relies on a novel inlier scale estimator. The proposed inlier scale estimator

†1 Osaka University
†2 Osaka Institute of Technology

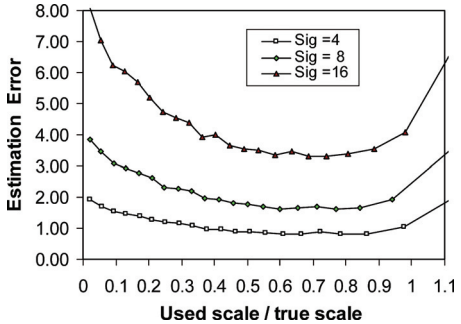


Fig. 1 Estimation error curves in simulation for line fitting with different Gaussian noise levels. The graph shows the statistical relation between the estimation error and the ratio between the inlier scale used and the true inlier scale. The results confirm that a robust estimator should use an inlier scale estimate as close to the true inlier scale as possible. Too small or too large an inlier scale estimate results in less robustness.

does not detect the inliers directly from the roughly estimated density using a smoothing parameter. Instead, we apply a matching method to detect inliers by globally searching the estimated density of the residual to find the most likely inlier residual distribution. The inlier residual distribution is modeled using a case-dependent but known constraint, the residual function, which importantly constrains the inlier residual distribution. This has not been used in any previous works. In our method, the inlier scale is estimated correctly, thus improving the estimation robustness.

The structure of our paper is as follows. First, we briefly review previous works in Section 2, and then we give an overview of the proposed method in Section 3. Section 4 states the definitions used in the paper. In Section 5, we carry out a residual distribution analysis for several estimation problems. In Section 6, we introduce the proposed inlier scale estimate based on the analysis of the residual distribution and describe the objective function to evaluate the estimate. Thereafter, various experiments are presented in Section 7 and the results compared with several popular robust estimators. Finally, we give our conclusions.

2. Related Works

The least squares (LS) method [1] is a simple basic method for parameter estimation. It has been extended in the M-estimators [1] by replacing the square function of the LS by a flexible symmetric function with a unique minimum at zero. The drawback of the simple LS method and M-estimators is the very low breakdown point. Improved algorithms for the LS method and M-estimators are available such as the least median squares (LMS) or reweighted LS, reweighted and redescending M-estimators [1], and these can achieve a higher breakdown point, up to 50% of the outliers. However, in a real estimation problem, such as extracting lines from an intensity image or extracting planes from a range image, where the outlier rate is much higher than 50%, the LS method and the M-estimators cannot function properly. Another drawback of the LS method and M-estimators is the initialization: as a result of improper initialization, the global minimum may not be obtained. This problem can however, be solved effectively using random sampling, as is the case in the well-known solutions LMedS, MSAC [17] and MLESAC [13].

Some estimators can tolerate higher outlier-rate than 50%. The RANSAC [2] and Hough transform [10] are the most popular in this category. If the scale of inliers is supplied, RANSAC can reach a very high breakdown point. However, the drawback of RANSAC and its subsequent improvements [11)–14)] is that they need a user-defined threshold to distinguish inliers. The Hough transform can also achieve a very high breakdown point so long as it is able to manage its large voting space. Certain extensions of LMS, such as MUSE (minimum unbiased scale estimate) [3] or ALKS (adaptive least kth order squares) [4], can be applied with high outlier rates, however these have a problem with extreme cases, such as those with very low or high outlier rates, and are sensitive to small pseudo structures. Another extension of LMS is MINPRAN (minimize probability of randomness) [5], which makes an assumption of the outlier distribution. This assumption seems to be strict since outlier distribution is assumed with difficulty. RESC (residual consensus) [6] computes a histogram of the residuals, then uses several parameters to compress the histogram, and finally the histogram power is computed as the score for the putative estimate. It is claimed that RESC can

tolerate single structure data containing up to 80% outliers, however, it needs many user-defined parameters to compress the histogram and to detect the inlier residual distribution, which reduces its adaptiveness. The pbM (projection-based M-estimator) [7),15),16)] is an extension of the M-estimator that uses projection pursuit and kernel density estimation (KDE), and can provide a breakdown point greater than 50%. However, it only works for linear (or linearized) residual functions, such as in linear regression. Another robust estimator that uses KDE is ASSC (adaptive scale sample consensus) [8)]. 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 for the KDE. ASSC has subsequently been improved as ASKC (adaptive scale kernel consensus) [9)]. ASKC improves the objective function of ASSC and the robustness in the case of a high outlier rate. However, in our experiments, ASKC and ASSC usually underestimated the population of inliers. The estimated inlier scale for these estimators correlates with their KDE bandwidth. Therefore, the objective function does not evaluate the estimate precisely, thus reducing the robustness of the estimators. More information about the robust estimators can be found in some reviews [17)–19)].

3. Overview of Proposed Estimator

In contrast to the pbM, ASSC or ASKC, our proposed method does not compute the inlier scale directly from the estimated residual density, since this only roughly describes the true distribution and the location of a local peak, global peak or local valley in the density estimation depends on a smoothing parameter (bandwidth or binwidth). We estimate the inlier bound by globally searching the inlier scale estimate that results in the best fit of the residual density to a residual distribution model. The low density tail of the residual distribution model is not used for the fitting and is assumed to come from outliers.

In previous methods, the residual distribution of inliers was typically assumed to be a Gaussian distribution. In our method, we carefully analyze the distribution of inliers using the residual function which constrains the distribution of residuals. The residual distribution model is determined statistically or mathematically using the residual function. This means that the distribution model of inlier residuals varies when we apply different residual functions. This analysis helps the proposed estimator correctly estimate the inlier scale, thereby improving the robustness.

4. Preliminaries

In this section, we describe the estimation problem and some definitions that are used in the paper.

Assume the estimation of a structure model with the constraint:

$$g(\boldsymbol{\theta}, \boldsymbol{X}) = 0, \tag{1}$$

where $\boldsymbol{\theta}$ is the parameter vector of the structure, and $\boldsymbol{X}$ is an explanatory data point. Our estimation problem is then described as:

- *Input*: N observed data points $\boldsymbol{X_i}, i = 1 \dots N$, including both inliers and outliers.
- *Output*: Parameter $\boldsymbol{\theta}$ that describes the data.

In a real problem, each inlier $\boldsymbol{X^t}$ is affected by an unknown amount of noise $\boldsymbol{n}$:

$$\boldsymbol{X} = \boldsymbol{X^t} + \boldsymbol{n}. \tag{2}$$

Therefore, the actual parameters $\boldsymbol{\theta}$ cannot be recovered, and some approximation of $\boldsymbol{\theta}$ needs to be estimated. A robust estimator based on random sampling like RANSAC solves the problem by trying many random trial estimates $\hat{\boldsymbol{\theta}}$, with the best estimate $\hat{\boldsymbol{\theta}}^*$ being the approximation of $\boldsymbol{\theta}$. In evaluating whether an estimate $\hat{\boldsymbol{\theta}}$ is good or bad, the estimator can only rely on the statistics of the error for each data point; this error is called the residual, which is a non-negative measure in the proposed method. For each model estimation problem, there are numerous ways of defining the residual function, including using the original constraint function (1). Generally, however, the residual is defined as:

$$r_{\hat{\theta}} = f(\hat{\boldsymbol{\theta}}, \boldsymbol{X}). \tag{3}$$

A good definition of the residual is that proposed by Luong, et al. [21)]:

$$r_{\hat{\theta}} = \frac{g(\hat{\boldsymbol{\theta}}, \boldsymbol{X})}{\| \bigtriangledown g(\hat{\boldsymbol{\theta}}, \boldsymbol{X}) \|}, \tag{4}$$

where $\bigtriangledown g(\hat{\boldsymbol{\theta}}, \boldsymbol{X})$ is the gradient of g with respect to variable $\boldsymbol{X}$.

In a real problem, the inlier residual is not zero. The standard deviation of these

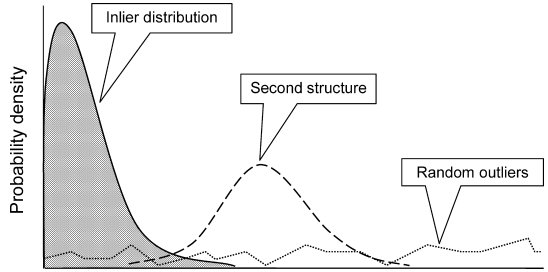


Fig. 2 Decomposition of residual density distribution: inlier density distribution and outlier (other) density distributions. The outlier distribution may consist of a distribution of the other structure and a distribution of random outliers.

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, an inlier scale estimator tries to estimate it. This estimate is denoted by $\sigma^*_{\hat{\theta}}$. Once the inlier scale has been found, the threshold $t_{\hat{\theta}} = \tau\sigma^*_{\hat{\theta}}$ can be decided to distinguish inliers from outliers.

Given an estimate $\hat{\boldsymbol{\theta}}$, and an inlier scale $\sigma_{\hat{\theta}}$, the probability density function for all residuals is denoted as $P_{\hat{\theta}}(r)$, which is the sum of density functions for inliers and outliers. The proposed estimator works with data with multiple structures, and therefore the residual distribution may have multiple modes. A segment of the distribution that has a mode near the origin is assumed to belong to the inlier structure, whereas the others belong to the outlier structures. The decomposition of the residual distribution is illustrated in **Fig. 2**. The outlier distribution is usually complicated and unpredictable. However, the inlier distribution can be well modeled in most problems. In our method, the inlier distribution model is made using the residual function. The density function for the standardized distribution model (SDM), with the sample deviation of 1, is denoted as $P(\xi)$, $\xi \geq 0$. Then, the inlier distribution is estimated by matching the residual distribution $P_{\hat{\theta}}(r)$ with SDM. Since the tail, with low density, of the inlier distribution is usually heavily overlapped with the outlier distribution, we do not use the whole SDM for matching. Only the dense segment of $P(\xi)$ with $0 \leq \xi \leq \kappa$ that contains most of population of SDM is used for matching. κ is selected so that the range $0 \leq \xi \leq \kappa$ contains more than 97% of the population. For example, when the SDM is the standard Gaussian distribution, we set $\kappa = 2.5$. In the following sections, the SDM is analyzed and modeled using the residual function.

5. Residual Distribution Analysis

In this section, we briefly describe the preliminaries associated with the estimation problem and then carry out an analysis of the residual distribution for various estimation problems. It is better to assume Gaussian noise on the data points than to assume a Gaussian distribution of residuals since the residual distribution is constrained by the residual function. This is because the noise on data points originates from physical sensors such as a camera in which noise distribution is usually modeled by a Gaussian distribution. Therefore, we assume that the noise model for the data points is known and, in this paper, is a Gaussian of unknown variance. However, due to the residual function (3), the distribution of residuals is generally different from that noise distribution. Then, we analyze the distribution model for residuals. Two examples are presented in this section: line fitting and fundamental matrix estimation.

5.1 Linear Residual

We start the analysis with a well-known problem for a robust estimator, the line fitting problem, in which the residual function is a linear function of the parameters. We have a set of N points (x, y), and the parameters of the true line l are slope (a, b) and intercept c, where a and b are normalized so that $a^2+b^2 = 1$. We denote these parameters as $\boldsymbol{\theta} = (a, b, c)$. In most computer vision problems, the data points are limited within some bound. Inliers are contaminated by noise with a noise model such that:

$$\begin{aligned} x &= x^t + n_x, \\ y &= y^t + n_y, \end{aligned} \tag{5}$$

where (x^t, y^t) is the true point and (n_x, n_y) is noise added to the point. The noise scale is assumed to be much smaller than the bound of the data points.

Given an estimate for the estimation of the line fitting problem: $\hat{\boldsymbol{\theta}} = (\hat{a}, \hat{b}, \hat{c})$, where $\hat{a}^2 + \hat{b}^2 = 1$, the fit of this estimate to the data set is analyzed by the residuals of all points. We focus on the analysis of the distribution of residuals. Signed residual r for data point (x, y) is computed as:

$$r = \hat{a}x + \hat{b}y + \hat{c}. \tag{6}$$

This is actually the signed point-line distance from (x, y) to the estimated line. For outliers, regardless of whether the estimate $\hat{\boldsymbol{\theta}}$ is correct or not, the residual r is still large and is bounded by the same limit $[r_{min}, r_{max}]$.

For inliers, r can be decomposed as follows.

$$\begin{aligned} r &= (\hat{a}x^t + \hat{b}y^t + \hat{c}) + (\hat{a}n_x + \hat{b}n_y) \\ &= r^t + r^n, \end{aligned} \tag{7}$$

where $r^t = \hat{a}x^t + \hat{b}y^t + \hat{c}$ and $r^n = \hat{a}n_x + \hat{b}n_y$. It can be seen that r is the sum of two different variables with different properties. r^t is the linear combination of x^t and y^t given the estimation parameters $\hat{a}, \hat{b}, \hat{c}$, and depends strictly on the accuracy of the estimation. r^n is the linear combination of the noise on the data points. If the noise on the data points is Gaussian noise, with some standard deviation and zero mean, $n_x \in G(\sigma_x, 0), n_y \in G(\sigma_y, 0)$. Then r^n is also a variable that comes from a Gaussian with standard deviation $\sigma_n = \sqrt{\hat{a}^2\sigma_x^2 + \hat{b}^2\sigma_y^2}$ and is bounded $\sigma_n < \sqrt{2(\sigma_x^2 + \sigma_y^2)}$. r^n does not really depend on the accuracy of the estimation. The better the estimate, the smaller r^t becomes and in the ideal case when the estimate is perfect, r^t =0, and the distribution of $r = r^n$ is entirely a Gaussian distribution.

This analysis can also be extended to any multiple linear regression problem in which the residual is a linear function of the variables:

$$r = \sum_{k=1}^{p} \hat{a}_k x_k + \hat{a}_0, \tag{8}$$

where $\hat{a}_k$ is a parameter of the estimation, and $(x_1 \dots x_p)$ is a data point. As the estimate improves, so the distribution of inlier residuals matches the Gaussian distribution more closely. In this case, the residual distribution model is a Gaussian distribution. The SDM is then the standard Gaussian distribution for the absolute of the variable.

5.2 Non-Linear Residual

Similar to Section 5.1, in this section we analyze the problem when the residual is a non-linear function or general function (3) of a data point. In this case, it is difficult to analyze the distribution mathematically. However, such a function constrains the distribution of residuals helping us to analyze it statistically by simulation, and then the ideal distribution of the residuals can be modeled.

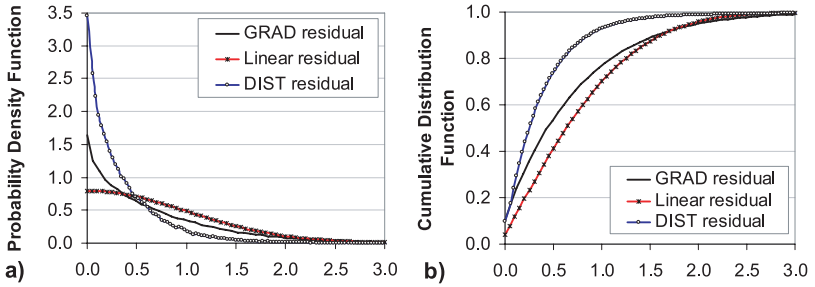


Fig. 3 Standardized Residual Distribution Model (SDM) of fundamental matrix estimation and line fitting problem with Gaussian noise on data points.

Implementation of this step can be done online.

Assuming a certain noise model on the data points, such as Gaussian noise on the data point $\boldsymbol{X}$, we can model how the residuals from inliers are distributed in the ideal case. In a complicated problem such as fundamental matrix estimation, it is easier to analyze by simulation. For a fundamental matrix estimation the constraint function of the data points is [20),21)]:

$$g(\boldsymbol{F}, \boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}'^{\boldsymbol{T}} \boldsymbol{F} \boldsymbol{x} = 0, \tag{9}$$

where $\boldsymbol{F}$ is the fundamental matrix and $\boldsymbol{X} = (\boldsymbol{x}, \boldsymbol{x}')$ is a single pair of point correspondences on two consecutive images. Several residual definitions exist, such as those in Ref. 21). Two non-linear residual functions are selected to simulate how the residuals are distributed.

- The first residual function, which is called GRAD in this paper, is based on a gradient criterion:

$$r = f(\boldsymbol{F}, \boldsymbol{x}, \boldsymbol{x}') = \frac{\left| \boldsymbol{x}'^{\boldsymbol{T}} \boldsymbol{F} \boldsymbol{x} \right|}{\sqrt{\| \boldsymbol{F}\boldsymbol{x} \|^2 + \| \boldsymbol{F}^T \boldsymbol{x}' \|^2}}. \tag{10}$$

- The second residual function, which is called DIST in this paper, uses symmetric distance from points to epipolar lines:

$$r = f(\boldsymbol{F}, \boldsymbol{x}, \boldsymbol{x}') = \left| \boldsymbol{x}'^{\boldsymbol{T}} \boldsymbol{F} \boldsymbol{x} \right| \sqrt{\frac{1}{\| \boldsymbol{F}\boldsymbol{x} \|^2} + \frac{1}{\| \boldsymbol{F}^T \boldsymbol{x}' \|^2}}. \tag{11}$$

The simulation is performed with an exceptionally large number of data points,

and the statistical results are shown in **Fig. 3**. For the ideal case in this simulation, residuals are calculated with a known fundamental matrix, zero-mean Gaussian noise is assumed on data point $\boldsymbol{X}$, and no outliers appear. The distribution of residuals is standardized so that the sample standard deviation, denoted by σ, is 1. Figure 3 shows the standardized residual distributions together with the standard Gaussian distribution for comparison. For the distribution of GRAD residuals, about 97.7% of the population is found within the range 2.5σ, and about 99.9% of residuals within 5σ. For the distribution of DIST residuals, about 97.6% of the population is found within the range 1.5σ, and about 99.7% of residuals within 5σ. For the Gaussian distribution, 97% of the population are within 2.5σ.

6. Proposed Robust Estimator

In the previous sections, we analyzed the distribution of residuals for several estimation problems. The understanding of the residual distribution thus gained prompts us to propose an inlier scale estimator. Similar to previous diagnostic robust estimators, the proposed estimator consists of two components: an inlier scale estimator and an objective function. We also use random sampling for the search procedure as in RANSAC.

6.1 Inlier Scale Estimation by Matching Residual Distribution and Residual Distribution Model

The inlier scale is estimated by searching the best fit between a segment of the residual distribution and the SDM. The segment of the residual distribution used for matching starts from zero. Then, the residual scale of the first structure is detected regardless of the outlier structures. The fitting error between the density function $P_{\hat{\theta}}(\rho)$ with assumed inlier scale σ and the SDM density function $P(\frac{\rho}{\sigma})$ is:

$$e_{\hat{\theta}}(\sigma) = \min_{\mu} \int_0^{\kappa\sigma} \left(P_{\hat{\theta}}(\rho) - \mu P\left(\frac{\rho}{\sigma}\right)\right)^2 d\rho, \tag{12}$$

where μ is some scale of the SDM density function, ρ is the residual variable and κ indicates the part of the SDM used in the matching as discussed in Section 4. The minimization (12) with respect to μ is solved when it is assigned:

$$\mu = \frac{\int_0^{\kappa\sigma} P_{\hat{\theta}}(\rho) P(\frac{\rho}{\sigma})\, d\rho}{\int_0^{\kappa\sigma} P(\frac{\rho}{\sigma})^2\, d\rho}. \tag{13}$$

Then, the best scale of inlier residuals $\sigma^*_{\hat{\theta}}$ is estimated by searching the scale that gives the smallest fitting error. This is summarized as

$$\sigma^*_{\hat{\theta}} = \underset{\sigma}{argmin}\{e_{\hat{\theta}}(\sigma)\}. \tag{14}$$

Inliers are then distinguished using the threshold $t_{\hat{\theta}} = \kappa\sigma^*_{\hat{\theta}}$. The inlier scale $\sigma^*_{\hat{\theta}}$ is refined for later use in the objective function, by being replaced by the standard deviation of estimated inliers:

$$\hat{\sigma}^*_{\hat{\theta}} = \sqrt{\int_0^{t_{\hat{\theta}}} \rho^2 P_{\hat{\theta}}(\rho)\, d\rho}, \tag{15}$$

In our algorithm, we compute the probability density of the residual from an estimate $\hat{\theta}$ by applying the well-known histogram method, although the KDE can also be used. A histogram is simple and as residual sorting is not required, in contrast to most previous estimators, it can be computed with low computational cost. Then, (12) and (13) are converted into histogram-based form, ρ is replaced by the bin variable $b_i = ib_{\hat{\theta}}$, which is the location of the i^{th} bin, with $b_{\hat{\theta}}$ the bin-width. The refined inlier scale in (15) is replaced by the sample deviation of inlier residuals $r_i \le t_{\hat{\theta}}$. In addition, $P_{\hat{\theta}}(b_i)$ is the count of residuals belonging to the i^{th} bin. Searching for the best inlier scale $\sigma^*_{\hat{\theta}}$ and $t_{\hat{\theta}}$ is graphically depicted in **Fig. 4**.

6.2 Bin-width Selection

Bin-width is the size of a bin in the residual histogram mentioned in Section 6.1. In this section, we decide the bin-width to be used in our algorithm. Bin-width (or bandwidth in previous works) affects the smoothness of the density distribution and consequently influences the detection of local peak or valley. Setting the bin-width is usually a difficult problem for those methods that rely on the probability density of residuals. A bin-width that produces good smoothness of the density estimation is required in such situation, and a widely used bin-width [26] for robust estimators is:

$$b_{\hat{\theta}} = \left(\frac{243 \int_{-1}^{1} K(\zeta)^2 d\zeta}{35N(\int_{-1}^{1} \zeta^2 K(\zeta) d\zeta)^2} \right)^{\frac{1}{5}} \hat{s}_{\hat{\theta}}, \tag{16}$$

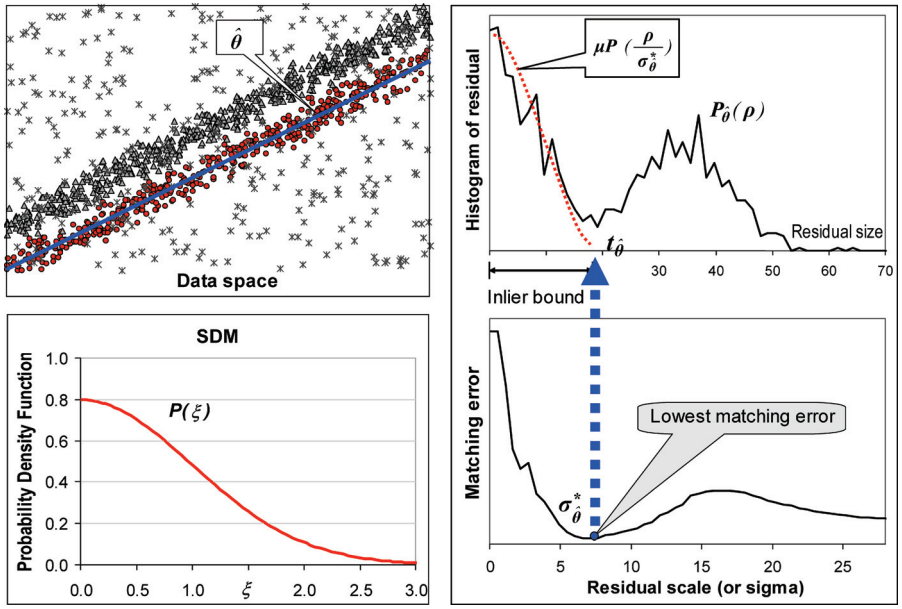


Fig. 4 Demonstration of finding the inlier bound. Data contains two parallel lines, and the SDM in this case is the Gaussian. The residual histogram is computed given the estimate $\hat{\theta}$, which has two actual modes for the two lines. The inlier scale is obtained by finding the smallest fitting error, and then the inlier bound is computed as $t_{\hat{\theta}} = \kappa\sigma^*_{\hat{\theta}}$.

where K is some kernel, such as the popular Gaussian kernel or the Epanechnikov kernel, $\hat{s}_{\hat{\theta}}$ is some scale estimate, such as the standard deviation of residuals, median scale estimate [1] or MAD estimate [1], and N is the number of data points. In our method, $\hat{s}_{\hat{\theta}}$ is the smallest window containing 15% of the smallest residuals.

Having obtained the bin-width, a histogram of the estimate can be built. Since the bin-width is small for outlier residuals, especially in case of high outlier-rates, the number of bins may be large and therefore, large number of bins for outliers should be ignored. For a specific unimodal distribution with deviation σ of N residuals, the bin-width is computed by (16), and by the definition, the densest bin contains fixed number of residuals, some scale of $0.15N$. Then, the number of bins for $r \leq \kappa\sigma$ is known, which is many less than N even when the distribution becomes a uniform distribution. However, if there are more than two component distributions (an inlier distribution and some outlier distributions), the number of bins may be many greater than N due to the large scale of outlier residuals. In order to search only for the inlier distribution, in practice, we limit the number of bins, for example, by N.

6.3 Objective Function

Inspired by the usage of the KDE in the pbM-Estimator [15),16)] and ASKC [9)], we also apply it in our adaptive objective function:

$$F(\hat{\boldsymbol{\theta}}) = \frac{1}{Nh_{\hat{\theta}}} \sum_{i=1}^{N} K\left(\frac{r_{i,\hat{\theta}}}{h_{\hat{\theta}}}\right), \tag{17}$$

where $h_{\hat{\theta}}$ is adaptively estimated and K is a kernel, such as the Gaussian kernel K_G or Epanechnikov kernel K_E. The KDE objective function evaluates how densely the residuals are distributed at zero using the kernel's window. In our case, the window of kernel K is $h_{\hat{\theta}}$, which tightly fits the estimated inliers, and therefore, the objective function gives the density measured at zero only for the estimated inliers. For K_G, $h_{\hat{\theta}} = \hat{\sigma}^*_{\hat{\theta}}$, while for K_E, $h_{\hat{\theta}} = \kappa\hat{\sigma}^*_{\hat{\theta}}$.

6.4 Estimation Algorithm Summary

A summary of the proposed algorithm is given below.

(a) Make the standardized residual distribution model (SDM) using the residual function. This can be done online or offline.
(b) Create a random sample and then estimate the putative parameters $\hat{\boldsymbol{\theta}}$.
(c) Estimate all the residuals of the data points given the parameters $\hat{\boldsymbol{\theta}}$.
(d) Estimate the bin-width as described in Section 6.2, and then compute the residual histogram $P_{\hat{\theta}}$.
(e) Estimate the inlier scale according to (14) and its refinement according to (15).
(f) Estimate the score using the objective function (17).
(g) Update the best solution.
(h) Repeat from (b) if not terminated.

The criterion for terminating the random sampling depends on the applications. It can be the excess of an amount of running time, or a number of iterations that assures a good estimate [23)]. In our experiments, we fix the same number of iterations for the proposed method as well as the compared methods.

7. Experiments

In this section, we describe the experiments carried out to validate our algorithm in both linear and non-linear estimation problems: plane fitting, line fitting and fundamental matrix estimation. For each problem, a simulation is first used to understand the various aspects of the algorithm and then a real experiment with real data is carried out to validate the algorithm in a real situation. For the plane and line fitting problems, we compared our algorithm with several popular robust estimators: the pbM-Estimator, LMedS, ALKS, ASSC, and ASKC. For the fundamental matrix estimation, we used LMedS, ASSC, ASKC, and ALKS for comparison since the pbM-Estimator was originally proposed for linear robust regression problems only. In the experiments using ALKS, since it is very unstable when the normalized error function accumulates only small number of residuals, we started using this error function only when it accumulated a number of residuals greater than 15% of the total number of data points. For the pbM-Estimator, we used the program from the authors [27]. The Epanechnikov kernel was used for all kernel density estimations including the related objective functions such as in the proposed objective function. All algorithms were supplied with the same set of random sampling trial hypotheses and no estimation optimization was done in any of the algorithms. In the proposed estimator, the value of κ is chosen according to the SDM. κ is selected so that the section of SDM for matching contains about 97% of the population. In the experiments, $\kappa = 2.5$ for the line fitting problem and fundamental matrix estimation using the GRAD function, while $\kappa = 1.5$ for the fundamental matrix estimation using the DIST function. The criteria for validating the proposed estimator are:

- robustness with various outlier rates and noise scales,
- accuracy of the inlier bound (threshold to distinguish the inliers), and
- the ability to work with data with multiple structures.

In data with the appearance of multiple structures, it is important that an estimator estimates a tight bound and outputs as many inliers as possible for a particular structure, otherwise the actual structure may be broken into many smaller structures or several structures may be estimated as a single one.

7.1 Linear Residual

In this problem, the estimator must extract the correct line or plane from a data set that contains single or multiple structures with the appearance of random outliers. The experiments were carried out by various popular and analytic simulations for a robust estimator as previous works. For data with a single structure, the evaluation was carried out with various outlier rates and noise scales. For data with multiple structures, we validated the proposed estimator using the various types of data with multiple structures frequently used for testing robust estimators: that is, data with parallel lines, data with steps and roof data.

Given an estimate $\hat{\boldsymbol{\theta}} = (\hat{a}, \hat{b}, \hat{c}, \hat{d})$, the residual function is defined as:

$$r_i = |\hat{a}x_i + \hat{b}y_i + \hat{c}z_i + \hat{d}|, \tag{18}$$

where (x_i, y_i, z_i) is a data point. The estimation error is defined as follows.

$$Error_{\hat{\theta}} = \sqrt{(a-\hat{a})^2 + (b-\hat{b})^2 + (c-\hat{c})^2 + (d-\hat{d})^2}, \tag{19}$$

where (a, b, c, d) are ground-truth parameters. The normal vector of each plane is normalized so that $\sqrt{a^2+b^2+c^2} = 1$, $\sqrt{\hat{a}^2+\hat{b}^2+\hat{c}^2} = 1$.

7.1.1 Single Structure with Various Outlier Rates

A 3D plane with 500 points was randomly generated for each trial data set. Gaussian noise with a mean of zero and noise scale σ_G was added to the inliers. Random outliers were generated to replace inliers, and therefore, the total data set always contained 500 points. All the points were located within the 3D volume [0, 0, 0, 1000, 1000, 1000]. 100 data sets were randomly generated, and for each data set, the same 10,000 iterations of random sampling were supplied to each estimator. The graphs shown below use the averages of the results for all 100 data sets.

We evaluated both the estimation error and inlier bound with various outlier rates. The ratio between the number of estimated inliers and the number of true inliers, and the ratio between the scale of the estimated inlier residual and the scale of the true inlier residual should be about 1 for any estimator.

In the first experiment for 3D plane fitting, we tested the outlier rate factor for all estimators with the same noise scale $\sigma_G = 8$. The average results are shown in **Fig. 5**. Figure 5. a) describes the break-down point and the accuracy

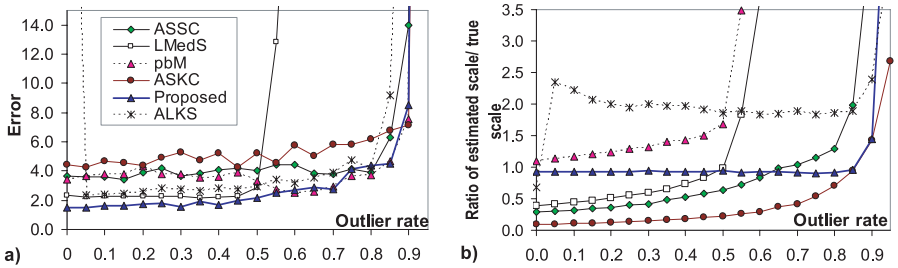


Fig. 5 Experiments with varying outlier rates for single-line data: a) estimation error, b) ratio between the scale of the estimated inlier residuals and the scale of residuals of true inliers.

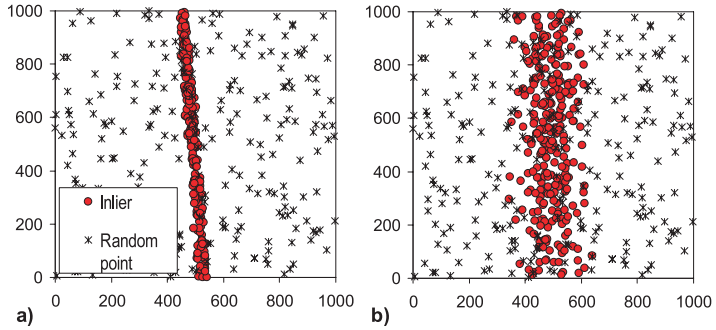


Fig. 6 Random data sets with an outlier rate of 60% and a) $\sigma_G = 8.0$ and b) $\sigma_G = 50$.

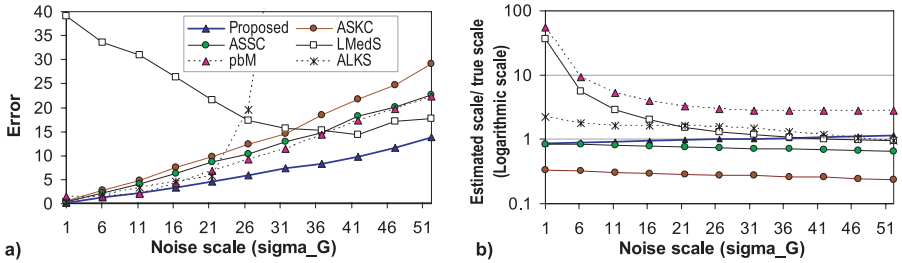


Fig. 7 Experiments with varying Gaussian noise scales and outlier rate fixed at 60%. Proposed estimator is highly resistant to high noise levels.

of the robust estimators, while Fig. 5. b) shows the ratio between the estimated and true inlier scales. We can see that in this experiment our proposed algorithm yields the best overall performance for accuracy and estimated inlier scale of all the algorithms. At low outlier rates, less than 50%, LMedS is accurate, but for higher outlier rates, LMedS fails to estimate. The performance of ALKS is unstable for very low or high outlier rates; the estimated inlier scale ratio is about 2, which means that ALKS overestimates the inlier scale. ASSC, ASKC, pbM and the proposed algorithm have similar breakdown points allowing these to retain good performance up to an outlier rate of 90%. ASSC and ASKC show similar performance, since their estimated inlier scales and KDE bandwidths correlate, but they usually underestimate the inlier scale. On the contrary, the performance of the pbM and proposed estimator for estimating residual density does not really depend on the bandwidth (or binwidth), and thus the accuracy of the pbM and proposed estimator remains high for the various outlier rates. In addition, as the proposed estimator always estimates an accurate inlier scale, the estimated inlier scale closely matches the true inlier scale. However, it should also be noted that the pbM estimates the solution first and then estimates the inlier scale and consequently the inlier scale is not important for the accuracy of the estimated solution.

7.1.2 Single Structure with Varying Noise Levels on Inliers

A second experiment was carried out to test all estimators with various noise scales. The data was set up similar to the experiment for 2D line fitting, except that the Gaussian noise scale σ_G on inliers varied between 1 and 52, while the outlier rate was fixed at 60%. Examples of the noise scales are shown in **Fig. 6**, while the average results are shown in **Fig. 7**. Figure 7. a) describes the estimation error, while Fig. 7. b) describes the ratio between the estimated inlier scale and true inlier scale. Since the outlier rate is 60%, LMedS fails to estimate correctly, giving a much larger estimated number of inliers than the number of true inliers. The performance of ALKS is unstable with the higher noise levels on inliers. All the other estimators have lower accuracy with higher noise levels, although the proposed estimator gives the most robust performance. These results confirm that our proposed estimator has the best accuracy and robustness of all the estimators, and the estimated inlier bound is quite close to the ground-truth.

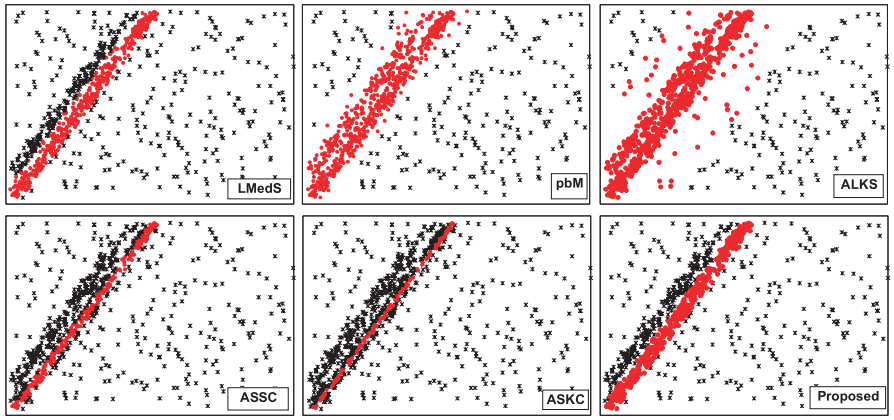


Fig. 8 Parallel Lines: estimation by each estimator using a random data set with $d = 70$. ALKS and the pbM are confused, since the two lines are extracted as one. ASKC and ASSC extract a small part of the actual line. LMedS estimates a line with a large number of inliers belonging to *Line* 1 and a few inliers belonging to *Line* 2. The proposed method extracts one of the two lines correctly and neatly.

7.1.3 Parallel Lines with Different Distances

Here we demonstrate the ability of the estimators with the appearance of multiple structures in the data.

A data set containing two parallel lines was used in this experiment. Each estimator was required to estimate one of the two lines correctly with a precise inlier bound. The experiment was carried out with different distances between the two parallel lines:

$Line\ 1 : 2x - y + d = 0,\ \ where\ d = 20, 30, 40, \ldots 210$

$Line\ 2 : 2x - y = 0.$

Various random data sets were used, with each data set containing 270 random outliers, 420 random points on *Line* 2, and 210 random points on *Line* 1. Gaussian noise $\sigma_G = 8.0$ was added to each point on each line, and the coordinates of all points were within the rectangle (0, 0, $62.5\,\sigma_G$, $62.5\,\sigma_G$). The estimations of the robust estimators using an example data set are shown in **Fig. 8**. In this example, all estimators estimated the correct line, but LMedS, the pbM and ALKS overestimated the population of inliers, ASSC and ASKC underestimated the

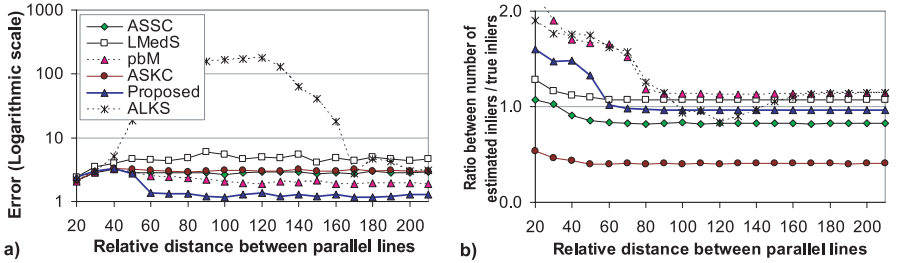


Fig. 9 Parallel Lines: a) estimation error, b) ratio between the number of estimated inliers and the number of true inliers.

inliers, while the proposed estimator estimated the inliers correctly. The average results for 100 random data sets are shown in **Fig. 9**. Figure 9. a) shows the estimation error for the robust estimators, while Fig. 9. b) shows the ratio between the number of estimated inliers and the number of true inliers. When the two lines are close together with $d = 20$, they are almost mistaken for being one line, with all estimators having a similar accuracy. When the lines are further apart, the performance of ALKS is the worst, as it only manages to estimate correctly once the two lines are very far apart with $d > 170$. This is understandable since it is claimed [4] that ALKS only estimates correctly step signals with a height greater than $8\,\sigma_G$. Because the actual outlier rate of estimating any line is greater than 50%, LMedS produces worse results as the two lines move further apart. ASSC and ASKC have a similar performance, but the number of inliers is underestimated in both cases and remains similar since it is only related to their KDE bandwidth. The proposed algorithm starts to estimate the line correctly for both solution parameters and inliers when $d = 60$, that is, when the distance between the lines is about $3.3\,\sigma_G$. With regards the bound on the estimated inliers, our proposed estimator gives the best results, since the number of estimated inliers is relatively close to the number of true inliers; in fact it is slightly smaller since leverage true inliers were also judged as outliers.

7.1.4 Multiple Structures: Steps with Varying Noise Levels

In this experiment, the step data consisted of four planes, set up as shown in **Fig. 10**. The parameters of the actual planes are:

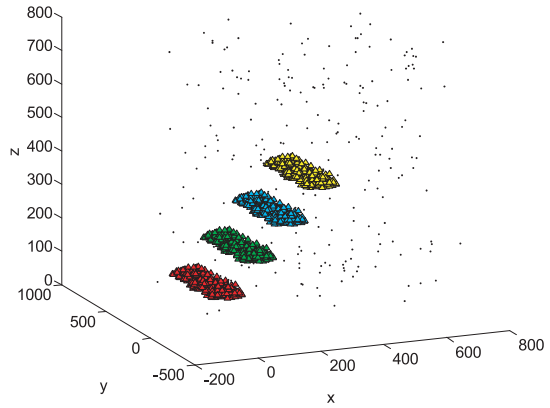


Fig. 10 An example of a step data set with four planes and various random points. Each point on a plane is contaminated by Gaussian noise with $\sigma_G = 5.0$.

$$Plane\ 1 : z - 100 = 0$$
$$Plane\ 2 : z - 200 = 0$$
$$Plane\ 3 : z - 300 = 0$$
$$Plane\ 4 : z - 400 = 0$$

The data set used in the evaluation consisted of 240 random points for each plane and 240 random outliers. Each data point on a plane was contaminated by Gaussian noise with σ_G. The experiment was carried out to test all the estimators with different values of σ_G. For larger values of σ_G, the four planes move closer and may become fused. The results are illustrated in **Fig. 11**, which gives the average of the results for 100 such randomly generated data sets.

In this experiment, the pbM-Estimator did not perform well since it mistook the four planes for the same structure, and consequently the estimated number of inliers is about four times the number of true inliers for each plane, as shown in Fig. 11. b). LMedS also did not perform adequately since the outlier rate is high for the estimation of any plane. ASSC and ASKC succeeded in estimating correctly with low noise levels only. The number of estimated inliers for the two methods remained similar regardless of whether they failed or succeeded. The proposed method was able to work correctly with slightly higher noise levels but

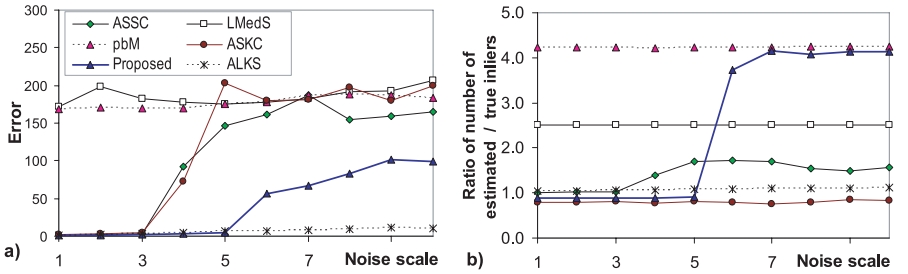


Fig. 11 Results from using data consisting of steps with different Gaussian noise levels $\sigma_G = 1, 2, \ldots, 10$. a) shows the average estimation error, while b) shows the ratio between the estimated number of inliers and number of true inliers.

then became confused, and the four planes were estimated as a single plane. In this comparison, ALKS worked correctly with much higher noise levels. However, since ALKS is well-known for its instability and sensitivity to small pseudo structures, we limited the size of possible structures, such that the estimated structure for ALKS was larger than 15% of the data. Therefore, it was able to estimate these steps correctly. In this case, its sensitivity was an advantage.

7.1.5 Multiple Structures: Roof with Varying Noise Levels

In this experiment, two planes were set up as shown in **Fig. 12**. The parameters of the actual planes are:

$$Plane\ 1 : x - y = 0$$
$$Plane\ 2 : x + y + 500 = 0$$

The data set used in the evaluation consisted of 350 random points for *Plane* 1, 350 random points for *Plane* 2 and 300 random outliers. Each data point on a plane was contaminated by Gaussian noise with σ_G. The experiment was carried out to test all the estimators with different values of σ_G. The results are depicted in **Fig. 13**, which gives the average results for 100 such randomly generated data sets.

The results show that most of the estimators worked well with this type of data except the LMedS since the actual outlier rate for estimating any plane was higher than 50%. The proposed estimator and pbM-Estimator outperformed the others. The pbM-Estimator performed slightly better with a low noise level, whereas the proposed estimator performed slightly better with high noise levels.

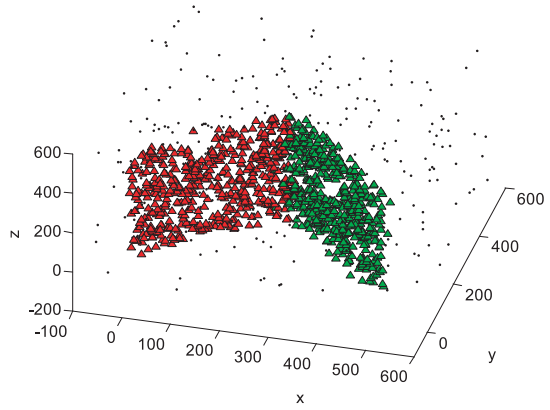


Fig. 12 An example of the roof data set consisting of two planes and various random points. Each point on a plane is contaminated by Gaussian noise with $\sigma_G = 13.0$.

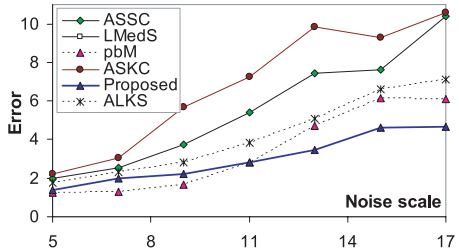


Fig. 13 Estimation error using roof data with different Gaussian noise levels $\sigma_G = 5, 7, \ldots, 17$.

The above results clearly show that our proposed algorithm works well with data containing multiple structures.

7.2 Non-linear Residual

For this experiment, we first carried out a simulation to validate various aspects of the proposed algorithm, and then performed the experiment with real data to show the effectiveness in a real situation. The GRAD and DIST residual definitions described in Section 5.2 were used for the fundamental matrix estimation. These residual definitions are not linear, and therefore, the pbM-Estimator is not applicable, because it was originally designed for linear residual problems only. Thus we compared the proposed algorithm with ASSC, ASKC, LMedS and ALKS, even though the non-linear residual function could have been linearized for use by the pbM.

Since it is not possible to compare the estimated fundamental matrix with a ground-truth fundamental matrix, we computed the error as the standard deviation of only the inlier residuals of the estimated fundamental matrix $\hat{\boldsymbol{\theta}}^* = \hat{\boldsymbol{F}}^*$:

$$Error_{\hat{\boldsymbol{F}}^*} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(r_{i,\hat{\boldsymbol{F}}^*})^2}, \tag{20}$$

where M is the number of inliers. This error computation relies on how the solution fits the motion data: a better fit produces smaller residuals for inliers, and vice versa. In the simulation, we know the true inliers and thus M is known. In the real experiment, the error is computed for the M smallest residuals (which are considered inliers), with M assigned manually after checking the actual data.

7.2.1 Fundamental Matrix Estimation in a Simulation

We simulated points on a unit sphere, with 500 points randomly distributed on a unit sphere. Altering the view point slightly causes the points on the sphere to move, thus creating 500 pairs of point correspondences. Some of these pairs were then replaced by outlying pairs with random point coordinates, thus keeping the total number of pairs as 500. Coordinates (x, y, z) for each inlier point on the unit sphere, before and after being moved, are contaminated by Gaussian noise with zero mean and noise scale σ_G. The fundamental matrix was estimated using the seven point algorithm [22]. Experiments were carried out for robustness under various outlier rates and the average results of 100 randomly generated data sets are shown in **Fig. 14** with $\sigma_G = 0.005$ for the GRAD residual function. The results of these experiments are similar to those in the plane fitting problem described above. These results prove that our proposed algorithm gives the highest robustness under various outlier rates and estimates a reasonable number of inliers which is close to the number of true inliers. ALKS performs quite well in these experiments and also produces an estimate or the number of inliers which is close to the number of true inliers, however it is unstable for very low or high outlier rates. ASSC and ASKC usually underestimate the population of inliers

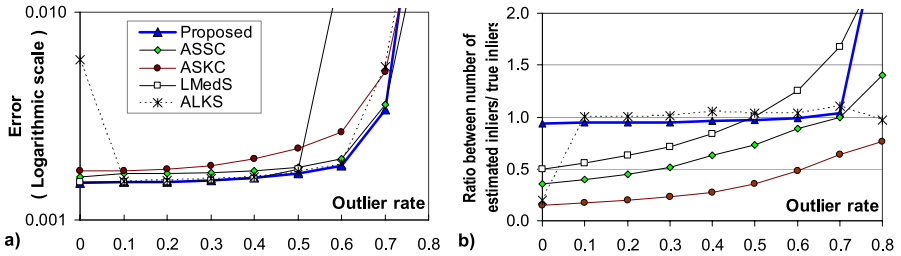


Fig. 14 Fundamental matrix estimation in a simulation using the GRAD residual function: a) estimation error, and b) ratio between the number of estimated inliers and the number of true inliers with various outlier rates.

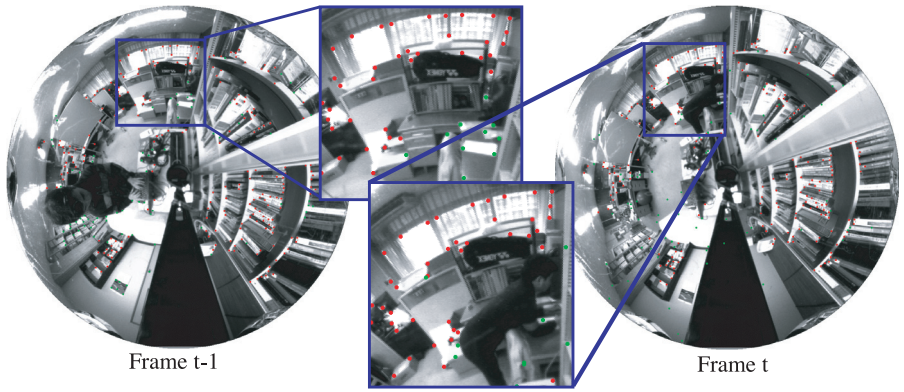


Fig. 15 A pair of images in a sequence: inliers (image features in red) and outliers (image features in green) are output by the proposed estimator.

as in the previous experiments.

We carried out a similar simulation using the DIST residual function. However, as the results are similar to those using the GRAD residual function they are not shown here.

7.2.2 Fundamental Matrix Estimation in Real Video Sequences

In this experiment, real video sequences were captured in an indoor environment with an omnidirectional vision sensor. Examples of the captured images are shown in **Fig. 15**. The sensor consisted of an omnidirectional mirror, a telecentric lens and an imaging sensor. The camera was mounted on a rotary stage and controlled by a PC, which translated the camera whilst it was being rotated. For each pair of images, 200 Harris image features were detected on the first image and tracked on the second image to obtain the feature correspondence pairs using the KLT feature tracker [25] implemented in OpenCV [24]. Features for each image were mapped to the unit sphere. The fundamental matrix between a pair of consecutive images was computed using the seven point algorithm with these feature correspondence pairs. For each video sequence, about 50 images were captured whilst ensuring the same rotation between consecutive images. The performance of all the estimators tends to deteriorate with a greater degree of rotation, since the KLT tracker is less accurate under greater rotation. Therefore, we used three video sequences with different rotation settings. These video sequences are referred to as *Video_4 deg*, *Video_14 deg* and *Video_18 deg* for rotation speeds of 4 degrees/frame, 14 degrees/frame, and 18 degrees/frame, respectively. We computed the error by (20) and M was set independently for each video sequence after randomly checking five pairs of images within each video sequence. The average number of true inliers and the assigned value of M for each video sequence are given in **Table 1**. From this table, we can see that the outlier rate for *Video_4 deg* is low, about 10%. For *Video_14 deg*, the outlier rate is about 50%, and for *Video_18 deg*, the outlier rate is about 65%. For each image pair, 20000 iterations of random sampling were provided for each estimator. In this case, the true noise model on the feature points was not known. However, it was assumed to be a Gaussian model with zero mean and thus the residual distribution models for the GRAD and DIST residual were known. In this experiment, the results for GRAD and DIST residual function are similar, the only description of experiment for GRAD is shown in this section.

The average error and number of estimated inliers for 100 executions of each video sequence are given in Table 1 and **Table 2**, respectively. The results show that the proposed estimator has the best accuracy for various outlier rates. The number of estimated inliers correlates with the outlier rate; it is slightly larger than the number of true inliers. ASSC and ASKC estimate a similar number of inliers for the various outlier rates as in the previous experiments. ALKS performs the worst of all these estimators in this real experiment.

In addition, we also evaluated the inlier bound for all estimators using visual-

Table 1 Fundamental matrix estimation for real video sequences using GRAD residual function: estimation error.

Video sequence	Video_4deg	Video_14deg	Video_18deg
Average number of true inliers	187.70/200	102.75/200	72.25/200
Assigned M	150	90	60
Fitting error			
Proposed method	0.000615	0.001493	0.001692
ASSC	0.000731	0.001673	0.001756
ASKC	0.000926	0.002350	0.002426
ALKS	0.004123	0.008205	0.008013
LMedS	0.000625	0.001676	0.002536

Table 2 Fundamental matrix estimation for real video sequences using GRAD residual function: number of estimated inliers.

Video sequence	Video_4deg	Video_14deg	Video_18deg
Average number of true inliers	187.70/200	102.75/200	72.25/200
Proposed method	182.307	110.847	87.935
ASSC	68.079	65.534	65.534
ASKC	23.122	24.073	24.073
ALKS	66.057	94.977	70.133
LMedS	101.000	101.000	101.000

ization. For each estimator, the average of all the histograms of residuals from the estimated solutions for all executions was calculated for comparison with the average of the estimated thresholds. This averaging for visualization purposes can be done within the same video sequence and with the same control speed only, since the performance of the KLT feature tracking is similar. The visualization is shown in **Fig. 16**. a), b) and c) for the three video sequences *Video_4 deg*, *Video_14 deg* and *Video_18 deg*, respectively. It can be seen from these figures that the proposed estimator output the most reasonable results, with the estimated threshold adaptively located in the tail of the actual distribution, and in which the density of outliers was low and the density of inliers within the inlier bound was high. This means that it was able to separate the inliers and outliers

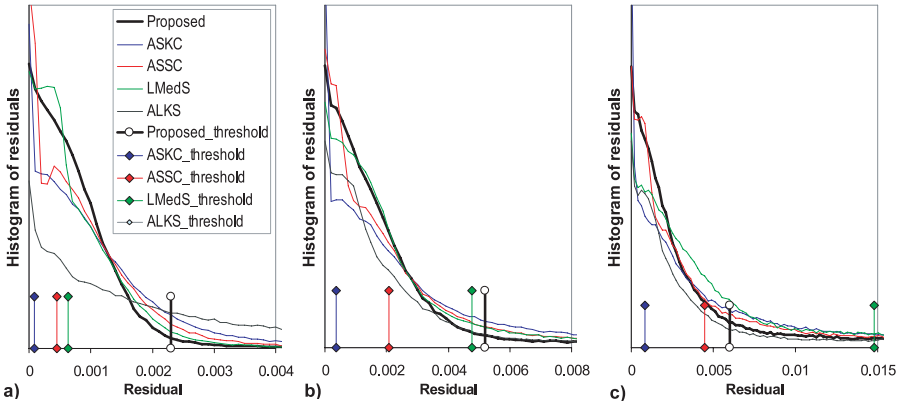


Fig. 16 Visualization of the estimated inlier bounds for estimators using the GRAD residual function with three videos sequences (from left to right) *Video_4 deg*, *Video_14 deg* and *Video_18 deg*. An average of the histograms of residuals from the estimated solutions was made for each estimator to visualize how tightly the estimated inlier bound fits the residual distribution.

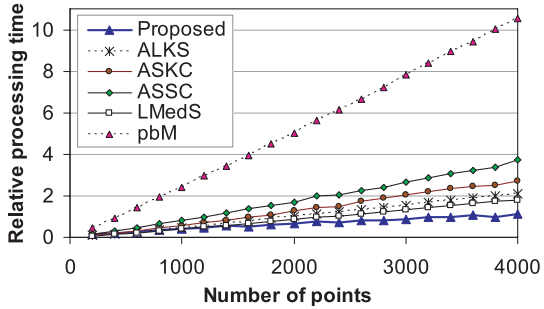


Fig. 17 Processing time for all estimators.

effectively. ASKC and ASSC output a solution in which the threshold was not located in the tail of the distribution, and in which the density of inliers was very high since the number of inliers was underestimated. ALKS did not work well resulting in a low density of histograms and very large inlier bounds. For the sake of giving only informative comparisons, the estimated inlier bounds for ALKS are not shown.

7.3 Computational cost

We simulated the relation between processing time and the number of data points, the average results of which are shown in **Fig. 17**. The graph shows that overall the proposed estimator gives the fastest computational time, especially for large data. For the proposed method, the residuals are not needed to be sorted, therefore it is fast in comparison with the others, especially when the number of data points increases. For the other estimators, the residuals have to be sorted first. LMedS is the simplest algorithm among the sorting-based methods, it takes the second fastest place in this comparison. After sorting the residuals, ALKS needs more cost to find the separation between inliers and outliers. ASKC and ASSC have the same procedure to locate the inlier distribution using mean-shift algorithm, the only difference is that ASKC uses the smaller window (bandwidth) for searching the local peaks of residual density then it consumes less computational cost than ASSC. The slowest estimator is pbM since it consumes heavy cost to find a global peak of residual density.

8. Conclusions

In this paper, we proposed a novel highly robust estimator for the estimation problem in computer vision that deals with data with high outlier rates and multiple structures. Our algorithm does not need any prior information about the inlier scale, as this is estimated adaptively. Depending on the specific problem, the distribution model of residuals is analyzed using that useful constraint, the residual function. The analysis is feasible and simple, and simulation of the residual distribution model can always be performed. The advantage of this approach is that it estimates the inlier scale correctly and therefore improves robustness. The proposed algorithm was positively validated through experiments with various conditions and real estimation problems. The use of the constraint from the residual function in the robust estimator is effective for improving the robustness and detection of inliers.

The proposed estimator can be applied to any problem in which the residual function is properly defined. Furthermore, it is especially useful when the inlier scale needs to be estimated accurately.

Acknowledgments This work was supported by the Grant-in-Aid for Scientific Research (S) (17100002).

References

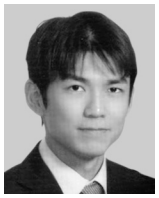
1) Rousseeuw, P.J. and Leroy, A.: *Robust Regression and Outlier Detection*. John Wiley & Sons, New York (1987).
2) Fischler, M.A. and Bolles, R.C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, *Comm. ACM*, 24, pp.381–395 (1981).
3) Miller, J.V. and Stewart, C.V.: MUSE: Robust surface fitting using unbiased scale estimates, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp.300–306 (1996).
4) Lee, K.M. Meer, P. and Park, R.H.: Robust adaptive segmentation of range images, *IEEE Trans. Pattern Anal. Machine Intelligence*, Vol.20, pp.200–205 (1998).
5) Stewart, C.V.: MINPRAN: A new robust estimator for computer vision, *IEEE Trans. Pattern Anal. Machine Intelligence*, Vol.17, pp.925–938 (1995).
6) Yu, X. Bui, T.D. and Krzyzak, A.: Robust Estimation for Range Image Segmentation and Reconstruction, *IEEE Trans. Pattern Anal. Machine Intelligence*, Vol.16, No.5, pp.530–538 (1994).
7) Chen, H. and Meer, P.: Robust regression with projection based M-estimators, *9th Intl. Conf. on Computer Vision*, pp.878–885 (2003).
8) Wang, H. and Suter, D.: Robust adaptive-scale parametric model estimation for computer vision, *IEEE Trans. Pattern Anal. Machine Intelligence*, Vol.26, No.11, pp.1459–1474, 2004.
9) Wang, H. Mirota, D. Ishii, M. and Hager, G.D.: Robust Motion Estimation and Structure Recovery from Endoscopic Image Sequences With an Adaptive Scale Kernel Consensus Estimator, *Conference on Computer Vision and Pattern Recognition* (2008).
10) Illingworth, J. and Kittler, J.: A survey of the Hough transform, *Computer Vision, Graphics, and Image Processing* (*CVGIP*), Vol.44, pp.87–116 (1988).
11) Nister, D.: Preemptive RANSAC for live structure and motion estimation, *Ninth IEEE International Conference on Computer Vision*, pp.199 (2003).
12) Matas, J. and Chum, O.: Randomized RANSAC with Sequential Probability Ratio Test, *Proc. International Conference on Computer Vision*, Vol.2, pp.1727–1732 (2005).
13) Torr, P.H.S. and Zisserman, A.: MLESAC: A new robust estimator with application to estimating image geometry, *Computer Vision and Image Understanding*, Vol.78, pp.138–156 (2000).
14) Torr, P.H.S. and Murray, D.W.: Guided-MLESAC: Faster Image Transform Estimation by Using Matching Priors, *IEEE Trans. Pattern Anal. Machine Intelligence*, Vol.27, No.10, pp.1523–1535 (2005).
15) Rozenfeld, S. and Shimshoni, I.: The Modified pbM-Estimator Method and a Run-

time Analysis Technique for the RANSAC Family, *Conference on Computer Vision and Pattern Recognition*, pp.1113–1120 (2005).
16) Subbarao, R. and Meer, P.: Beyond RANSAC: User Independent Robust Regression, *Conference on Computer Vision and Pattern Recognition Workshop* (*CVPRW'06*), p.101 (2006).
17) Torr, P.H.S. and Murray, D.W.: The development and comparison of robust methods for estimating the fundamental matrix, *Int. J. Computer Vision*, Vol.24, pp.271–300 (1997).
18) Stewart, C.V.: Robust Parameter Estimation in Computer Vision, *SIAM Rev.*, Vol.41, Iss.3, pp.513–537 (1999).
19) Meer, P. Mintz, D. Rosenfeld, A. and Kim, D.Y.: Robust regression methods for computer vision: A review, *Int. J. Computer Vision*, Vol.6, No.1, pp.59–70, Apr. (1991).
20) Longuet-Higgins, H.C.: A computer algorithm for reconstructing a scene from two projections, Nature, Vol.293, pp.133–135 (1981).
21) Luong, Q.T. and Faugeras, O.D.: The fundamental matrix: Theory, algorithms, and stability analysis, *Int. J. Computer Vision*, Vol.17, No.1, pp.43–75 (1996).
22) Hartley, R.I.: Projective reconstruction and invariants from multiple images, *IEEE Trans. Pattern Anal. Machine Intelligence*, Vol.16, No.10, pp.1036–1041 (1994).
23) Forsyth, D.A. and Ponce, J.: Computer Vision: A Modern Approach, Prentice Hall (2002).
24) Intel Corporation, Open Source Computer Vision Library, http://www.intel.com/technology/computing/opencv/index.htm
25) Shi, J. and Tomasi, C.: Good Features to Track, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp.593–600 (1994).
26) Wand, M.P. and Jones, M.: *Kernel Smoothing*, Chapman & Hall (1995).
27) Subbarao, R. and Meer, P.: pbM-Estimator source code, http://www.caip.rutgers.edu/riul/research/robust.html

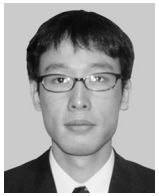
(Received April 22, 2009)
(Accepted September 2, 2009)
(Released November 16, 2009)

(Communicated by *Vincent Lepetit*)

Trung Thanh Ngo received his M.E. degree from Osaka University in 2005. He is currently a Ph.D. candidate at Osaka University and working as a Research Assistant in the Department of Intelligent Media, Institute of Scientific and Industrial Research, Osaka University.



Hajime Nagahara received his B.E. and M.E. degrees in Electrical and Electronic Engineering from Yamaguchi University in 1996 and 1998, respectively. He received his Ph.D. in System Engineering from Osaka University in 2001. He was a Research Associate of the Japan Society for the Promotion of Science in 2001–2003. He was a Research Associate of Graduate School of Engineering Science, Osaka University, in 2003–2006. He was a Visiting Associate Professor at CREA University of Picardie Jules Verns, France in 2005. He has been an Assistant Professor of Graduate School of Engineering Science since 2007. He was a Visiting Researcher at Columbia University, USA in 2007–2008. His research interests include image processing, computer vision and virtual reality. He received the ACM VRST2003 Honorable Mention Award in 2003.



Ryusuke Sagawa is an Assistant Professor at the Institute of Scientific and Industrial Research, Osaka University, Osaka, Japan. He received the B.E. in Information Science from Kyoto University, Kyoto, Japan, in 1998. He received the M.E. in Information Engineering in 2000 and the Ph.D. in Information and Communication Engineering from the University of Tokyo, Tokyo, Japan in 2003. His primary research interests are computer vision, computer graphics and robotics (mainly geometrical modeling and visualization). He is a member of IEICEJ, IPSJ, RSJ, and IEEE.

Yasuhiro Mukaigawa received his M.E. and Ph.D. degrees from University of Tsukuba in 1994 and 1997, respectively. He became a Research Associate at Okayama University in 1997, an Assistant Professor at University of Tsukuba in 2003, and an Associate Professor at Osaka University in 2004. His current research interests include photometric analysis and computational photography. He was awarded the MIRU Nagao Award in 2008. He is a member of IPS, VRSJ, and IEEE.

Masahiko Yachida received the B.E. and M.S. degrees in Electrical Engineering, and the Ph.D. in Control Engineering, all from Osaka University, Osaka, Japan in 1969, 1971, and 1976, respectively. He joined the Department of Control Engineering, Faculty of Engineering Science, Osaka University in 1971 as a Research Associate and became an Associate Professor at the same department. He then moved to the Department of Information and Computer Science as a Professor in 1990 and was a Professor at the Department of System Engineering at the same University. He was a Professor of Systems and Human Science, Graduate School of Engineering Science, Osaka University. He has been a Professor in the Faculty of Information Science and Technology, Osaka Institute of Technology since 2008. He is an author of Robot Vision (Shoko-do, received Ohkawa Publishing Prize), a co-author of Pattern Information Processing (Ohm-sha) and an editor of Computer Vision (Maruzen) and other books. He was the Chairman of Technical Committee on Computer Vision, Information Processing Society of Japan and the Chairman of Technical Committee on Pattern Recognition & Media Understanding, Institute of Electronics Information & Communication Engineers, Japan. His interests are in the field of computer vision, image processing, mobile robotics and artificial intelligence.

Yasushi Yagi is the Professor of Intelligent Media and Computer Science, and the Assistant Director of the Institute of Scientific and Industrial Research, Osaka University, Ibaraki, Japan. He received his Ph.D. degree from Osaka University in 1991. In 1985, he joined the Product Development Laboratory, Mitsubishi Electric Corporation, where he worked on robotics and inspections. He became a Research Associate in 1990, a Lecturer in 1993, an Associate Professor in 1996, and a Professor in 2003 at Osaka University. International conferences for which he served as chair include: FG1998 (Financial chair), OMINVIS2003 (Organizing chair), ROBIO2006 (Program co-chair), ACCV2007 (Program chair), PSVIT2009 (Financial chair) and ACCV2009 (General chair). He is the Editor of IEEE ICRA Conference Editorial Board (2007, 2008), the Editor-in-chief of IPSJ Transactions on Computer Vision & Image Media, and the Associate Editor-in-chief of IPSJ Transactions on Computer Vision & Applications. He was awarded the ACM VRST2003 Honorable Mention Award, IEEE ROBIO2006 Finalist of T.J. Tan Best Paper in Robotics, IEEE ICRA2008 Finalist for Best Vision Paper, and MIRU2008 Nagao Award. His research interests are computer vision, medical engineering and robotics. He is a member of IPS, RSJ, and IEEE.