Vol.2010-CVIM-172 No.38
2010/5/28

Outlier Detection for Robust Parameter Estimation Against Multi-modeled/structured Data

Ngo Trung Thanh,[†1] Hajime Nagahara,[†2]
Ryusuke Sagawa,[†3] Yasuhiro Mukaigawa,[†1]
Masahiko Yachida[†4] and Yasushi Yagi[†1]

Model parameter estimation and automatic outlier detection is a fundamental and important problem in computer vision. Vision data is noisy and usually contains multiple structures, models of interest. RANSAC has been proven to be the most popular and effective solution for such problem, but it requires some user-defined threshold to discriminate inliers/outliers. It is then improved by the adaptive-scale robust estimators, which do not require the user-defined threshold and detect inliers automatically. However, there still remains some problem. The problem is that these adaptive-scale robust estimators do not focus on the accurate inlier detection.

In this paper, we propose several adaptive-scale robust estimators which can detect inliers accurately. There are two reasons for the idea of accurate inlier detection. First, if a robust estimator detects inliers better, then the robustness of the estimation can be improved. Second, in many real applications such as motion segmentation and range image segmentation, if the inlier detection is not very well, then a structure can be broken into smaller structures, an under-segmentation problem, or united with the other structures, an over-segmentation problem.

In the experiments, various analytic simulations in many aspects have shown the advantage of the proposed robust estimators compared to several latest robust estimators. The real experiments were also performed to prove the validation of the proposed estimators in real applications.

1. Introduction

Vision supplies us with plenty of information and various computer vision algorithms have been being proposed to exploit. We can now extract the geometric, photometric, semantic information and so on from all types of image such as intensity, infrared or range image. The applications of computer vision may include the extracting the geometric primitives from images like lines, planes, surfaces of known mathematic model[1)], multiple view image transformation[3)], motion estimation of the camera that is attached to a robot or an autonomous vehicle[3),27)], camera parameter calibration[2)], image recognition[4)], searching for the existance of a pattern in the image database, and in varieties of problems where the model that describes the mathematical formulation is known but the parameters of the model are unknown.

Vision data like intensity or range image always contains large number of pixels and each of them is captured with unknown amount of noise caused by the sensor. Therefore, the computer vision algorithms usually have to work with the heavily over-constrained problem. Vision data usually contains several structures of the same model such as different lines of same 2D line model or different circles of the same 2D circle model. Moreover, multiple models may appear at the same time. The different situations of vision data are illustrated in **Fig. 1**.

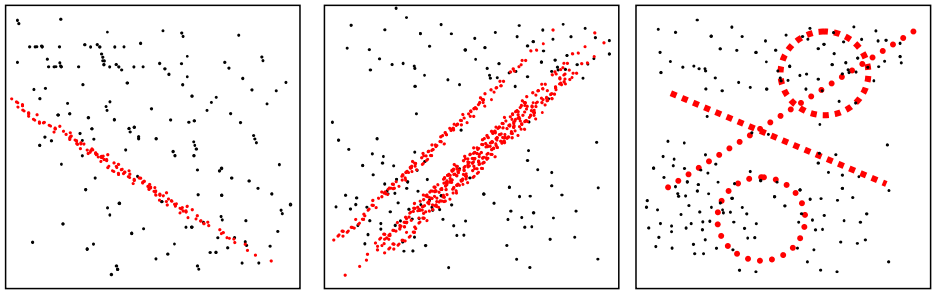

Fig. 1 Vision data may contain single structure, multiple structures of the same model, or multiple structures from multiple models and random noise. We should note that the data points for each structure are also contaminated by some small unknown source of noise.

Therefore, in most computer vision problems, to extract an interested struc-

†1 Osaka University
†2 Kyushu University
†3 Advanced Industrial Science and Technology
†4 Osaka Institute of Technology

ture, the parameter estimation methods must distinguish the data points of this structure from other structures and other models. RANSAC[6] that was proposed by M.A. Fischler and R.C. Bolles has been proven the most robust and effective method for such problems. Although, in computer vision, Hough transform or randomized Hough transform[15),16)] are also very robust estimators, they are only efficient for low dimensional parameter vector estimation. For a high dimensional parameter vector, it is is inefficient to manage the huge voting space.

In this paper, we present some new robust estimators, that rely on a novel inlier scale estimators. The proposed inlier scale estimators do 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 methods, the inlier scale is estimated correctly, thus improving the estimation robustness.

2. Related Works

The least squares (LS) method[5] is a simple basic method for parameter estimation. It has been extended in the M-estimators[5] 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[5], 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[23] and MLESAC[19].

Some estimators can tolerate higher outlier-rate than 50%. The RANSAC[6] and Hough transform[15] 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[17)18)19)20)] 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)[7] or ALKS (adaptive least kth order squares)[8], 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)[9], which makes an assumption of the outlier distribution. This assumption seems to be strict since outlier distribution is assumed with difficulty. RESC (residual consensus)[10] 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)[11)21)22)] 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)[12]. 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)[13]. 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.

3. Overview of Proposed Estimator

In contrast to the pbM, ASSC or ASKC, our proposed methods do 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.

In previous methods, the residual distribution of inliers was typically assumed to be a Gaussian distribution. In our methods, 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 estimators 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 $\boldsymbol{X_i}, i = 1..N$, 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 $\boldsymbol{n}$:

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

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.[25]:

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

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 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 estimators work 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 methods, 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.

An adaptive-scale robust estimator consists of two constituents[7)–10),12),14)]: a scale estimator and a hypothesis evaluator. For a given putative hypothesis, the scale estimator has to detect the inliers and estimate the inlier scale. Then

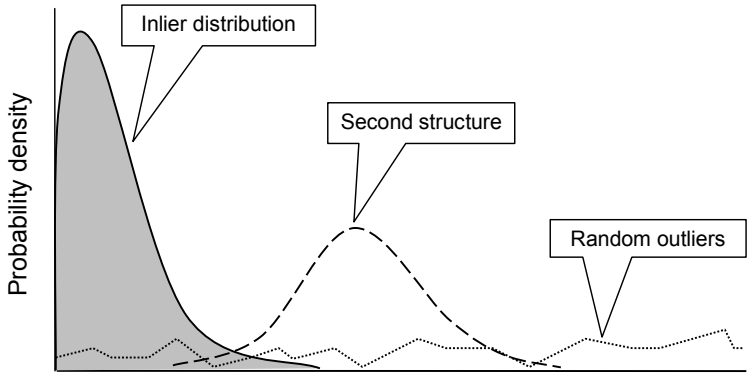

Fig. 2 Decomposition of residual density distribution: inlier density distribution and outlier (other) density distributions. The outlier distribution may consist of a distribution of the other structure and a distribution of random outliers.

the hypothesis evaluator computes the score using a objective function for this detection based on the estimated inlier scale. This is different from that of pbM[11),21),22)], in which inliers are detected finally after the solution is outputted.

The general flowchart for the proposed adaptive-scale robust estimators is presented in **Fig. 3**, the details are described in the following sections. The inlier residual distribution model, SDM, is described in Section 5. A histogram of residuals is computed using the bin-width that is discussed in Section 6.3. Searching for an inlier scale, inlier detection methods, is described in Section 6.1 and Section 6.2. And finally, the hypothesis is evaluated by an objective function in Section 6.4.

5. Case-dependent Residual Distribution Analysis

In most previous works[8),10),12)–14)], the theory of the algorithms is made for the Gaussian distribution of inlier residuals. However, it is not always true in practice. Therefore, we would like to formulate a residual distribution closer to the actual inlier residual distribution, in order to extract the inliers better. Our idea is that the distribution of inlier residuals depends on the residual function.

In this section, we carry out an analysis of the residual distribution for various estimation problems. It is better to assume Gaussian noise on the data points

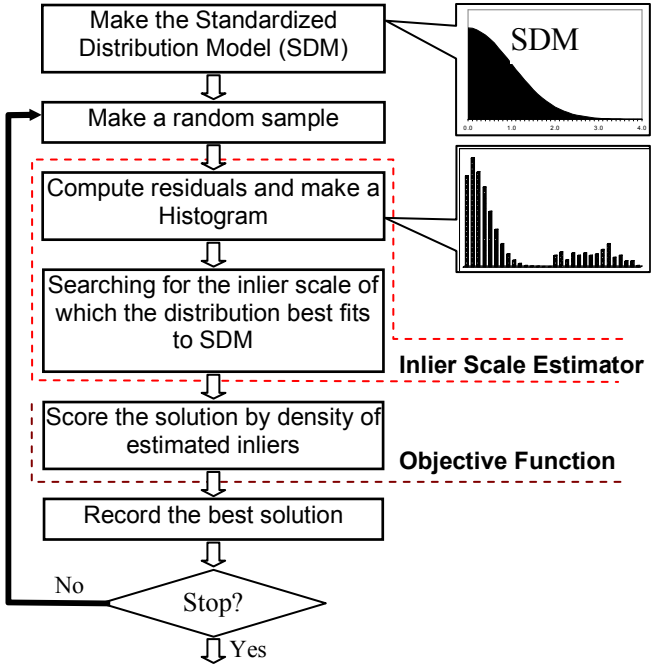

Fig. 3 The general flowchart for the proposed adaptive-scale robust estimators.

than to assume a Gaussian distribution of residuals. Firstly, 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. Secondly, it is because the residual distribution is constrained by the residual function. Therefore, we assume that the noise model for the data points is known and it is a Gaussian of unknown variance in this paper. 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 $(x_1..x_p)$ 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. 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[24)25)]:

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

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[25)]. 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 sym-

metric distance from points to epipolar lines:

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

The simulation is performed with an exceptionally large number of data points, and the statistical results are shown in **Fig. 4**. 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. Fig. 4 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.

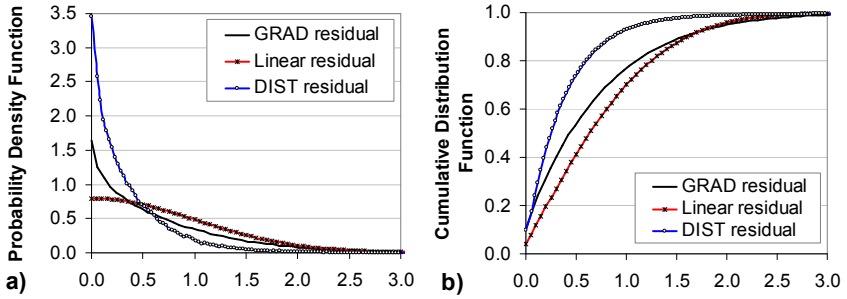

Fig. 4 Standardized Residual Distribution Model (SDM) of fundamental matrix estimation and line fitting problem with Gaussian noise on data points.

6. Inlier Detection Methods

In this section, we describe about the robust methods to detect inliers by fitting the actual residual distribution of putative hypothesis with the SDM to find the inlier distribution. The first method uses only a dense segment of SDM for fitting. This is fast but it requires some parameter, even thought it is easy to setup in practice. The second method does not limit the SDM and uses fitting the whole SDM. In this method, both inlier distribution parameter (inlier residual deviation) and also the average of ground distribution (additive outlier distribution). Which helps the second method works more robust though it is a bit slower.

6.1 Fitting for One Parameter

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 $\kappa = 2.5$.

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:

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

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:

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

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

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

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 $r_i \leq t_{\hat{\theta}}$. 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. 5**.

6.2 Fitting for Two Parameters

In the above method, we model the actual residual distribution as the separated inlier distribution and outlier distribution and use limited SDM for the fitting to find the inlier distribution. Even though, this algorithm works quick and quite well against the data with more than 80% of outlier rate in experiments, theoretically, this is true when the outlier distribution does not accommodate within the inlier distribution.

To realize the drawback of the above model for actual residual distribution, we propose another model for residual distribution. Residual distribution is assumed to consist of inlier distribution, outlier distribution and ground (outlier) distribution, which is illustrated in Fig. 2.

The fitting error between the density function $P_{\hat{\theta}}(\rho)$ with assumed inlier scale σ and the unlimited SDM density function $P(\frac{\rho}{\sigma})$ is:

$$e_{\hat{\theta}}(\sigma) = \min_{\mu,h} \int_0^{+\infty} \left(P_{\hat{\theta}}(\rho) - \mu P(\frac{\rho}{\sigma}) - h\right)^2 d\rho, \tag{16}$$

where h is the ground distribution which is added to inlier distribution. Compared to the model fitting error function (12), in (16), parameter h is to compensate for the high ground distribution. The minimization (16) can also be solved straightforwardly:

$$\mu = \frac{\int_0^{+\infty}(ab^2)d\rho \int_0^{+\infty} bd\rho - \int_0^{+\infty}(ab)d\rho \int_0^{+\infty}(b^2)d\rho}{\int_0^{+\infty}(b^3)d\rho \int_0^{+\infty}(b)d\rho - \left(\int_0^{+\infty}(b^2)d\rho\right)^2}, \tag{17}$$

and

$$h = \frac{\int_0^{+\infty}(b^3)d\rho \int_0^{+\infty}(ab)d\rho - \int_0^{+\infty}(ab^2)d\rho \int_0^{+\infty}(b^2)d\rho}{\int_0^{+\infty}(b^3)d\rho \int_0^{+\infty}(b)d\rho - \left(\int_0^{+\infty}(b^2)d\rho\right)^2}, \tag{18}$$

where

$$a = P_{\hat{\theta}}(\rho), b = P(\frac{\rho}{\sigma}).$$

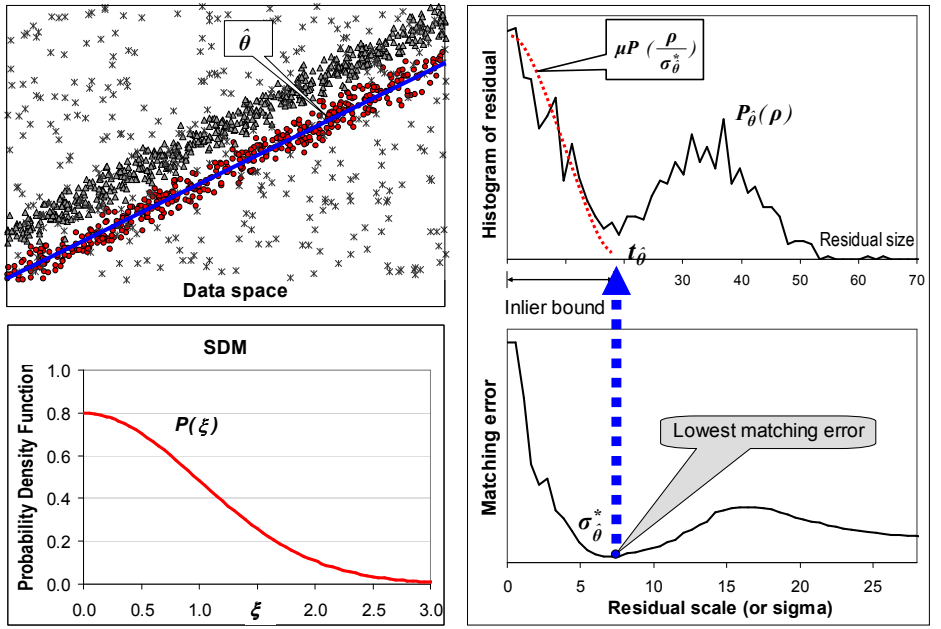

Fig. 5 Demonstration of finding the inlier bound. Data contains two parallel lines, and the SDM in this case is the Gaussian. The residual histogram is computed given the estimate $\hat{\theta}$, which has two actual modes for the two lines. The inlier scale is obtained by finding the smallest fitting error, and then the inlier bound is computed as $t_{\hat{\theta}} = \kappa\sigma^*_{\hat{\theta}}$.

Searching for the best inlier scale $\sigma^*_{\hat{\theta}}$ is graphically depicted in **Fig. 6**. We can clearly see that, the fitting error function $e_{\hat{\theta}}(\sigma)$ in this method is smoother than that of the previous method, which is promising to produce more robustness in practice.

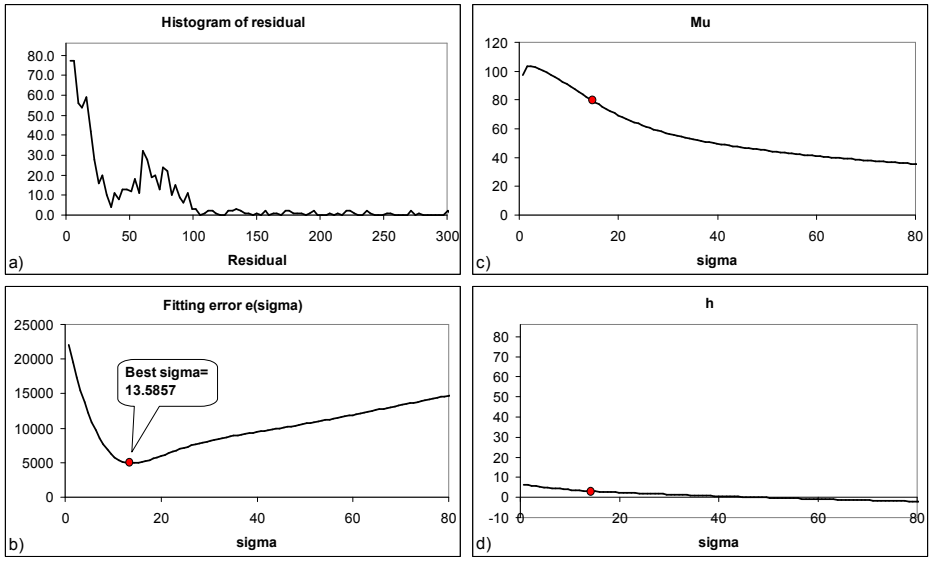

Fig. 6 Demonstration of finding the σ for inlier distribution. Data contains two parallel lines similar to that in Fig. 5, and the SDM in this case is the Gaussian. The residual histogram, ***a)***, is computed given the estimate $\hat{\theta}$, which has two actual modes for the two lines. The inlier scale is obtained by finding the smallest fitting error, ***b)***. The corresponding μ and h for the best σ, $\sigma^*_{\hat{\theta}}$, are given in ***c)*** and ***d)***, respectively.

6.3 Bin-width for Density Estimation

Bin-width (smoothing parameter or bandwidth for KDE) is the size of a bin in the residual histogram mentioned in Section 6. 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.

In this section, we describe about two choices for the smoothing parameter. The first is fixed and is widely used in robust estimators. It needs some experience to produce the smoothness of density estimation. The second smoothing parameter is totally data driven, and consequently it does not require any experience of user for all the situation.

6.3.1 Fixed Bin-width Computation

A bin-width that produces good smoothness of the density estimation is required in such situation, and a widely used bin-width[30] for robust estimators is:

$$b_{\hat{\theta}} = \left(\frac{243 \int_{-1}^{1} K(\zeta)^2 d\zeta}{35N(\int_{-1}^{1} \zeta^2 K(\zeta) d\zeta)^2} \right)^{\frac{1}{5}} \hat{s}_{\hat{\theta}}, \tag{19}$$

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[5] or MAD estimate[5], 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.

6.3.2 Adaptive Bin-width Computation

In the previous Section 6.3.1, we use a fast computation for bin-width similar to most of previous adaptive-scale robust estimators. It requires some experience of user for the smoothness of the computed density. It can not be adaptively changed for a dynamic situation such as when the outlier rate is varied while capturing the data. Therefore, an adaptive bin-width is desirable for a robust estimator. There exist number of previous works for binwidth computation[30)–34)], however these methods for computing the density is not suitable since they do not focus only on the inlier distribution.

In searching for a solution, we has become fascinated by the one of previous robust estimators, ALKS (adaptive least k^{th} order squares)[8] which is an improvement of MUSE (minimum unbiased scale estimator)[7].

Given an hypothesis $\hat{\theta}$, all the residual r_i for n data points are computed. In ALKS, all residuals are then sorted increasingly. ALKS proposes an *normalized*

error[8]):

$$\epsilon_k^2 = \frac{1}{q_k - p}\sum_{i=1}^{q_k}\left(\frac{r_i}{\hat{s}_k}\right)^2 = \frac{\hat{\sigma}_k^2}{\hat{s}_k^2}, \tag{20}$$

where p is the size of random sample for random sampling, $\hat{\sigma}_k^2$ is the standard deviation of first k residuals, $\hat{s}_k$ is unbiased scale estimate[7]):

$$\hat{s}_k = \frac{r_k}{\Phi^{-1}\left(0.5(1+\frac{k}{n+1})\right)}, \tag{21}$$

and $\Phi(u,1)$ is the cumulative density function for Gaussian distribution. $\Phi^{-1}(.)$ is the argument of the normal cumulative density function having the value inside the bracket, which is used as a compensation factor for (20). This compensation factor is just an approximation since in practice the data is contaminate with unknown proportion of outliers. q_k is the number of inliers decided by the threshold $2.5\hat{s}_k$. In our experience, ALKS has an ability to detect the inlier distribution quite well under the complicate multi-structural data. However, the inliers/outliers dichotomy is usually located outside the inlier distribution, which means ALKS usually over-estimates the inliers. Another problem for ALKS is that it needs the minimum size of a meaningful structure, however it is reasonable in practice where the size of data is perceivable. If there exists extreme outliers in the data, $\hat{s}_k$ can be come so large that minimum of the criterion is produced at an incorrect k due to the relative relation between inlier and outlier distributions.

Taking the advantage of the excellent ability to detect the structure, we solve the drawbacks of ALKS and use this algorithm not for a robust estimator but for computing an adaptive bin-width. We replace (20) by the following term:

$$\zeta_k^2 = \frac{1}{k - p}\sum_{i=1}^{k}\left(\frac{r_i}{r_k}\right)^2. \tag{22}$$

In this normalized error function, the compensation factor is removed for its ineffective performance. From(22), we can see that ζ_k is limited by 1: $0 < \zeta_k < 1$. We use this function not to find the inlier but to find most inliers to use in the bin-width computation (19). The algorithm is described as follows:

(1) Find the location k_{max} of global maximum ζ_{max} of ζ_k

$$k_{max} = \underset{p<k\leq n}{argmax}(\zeta_k).$$

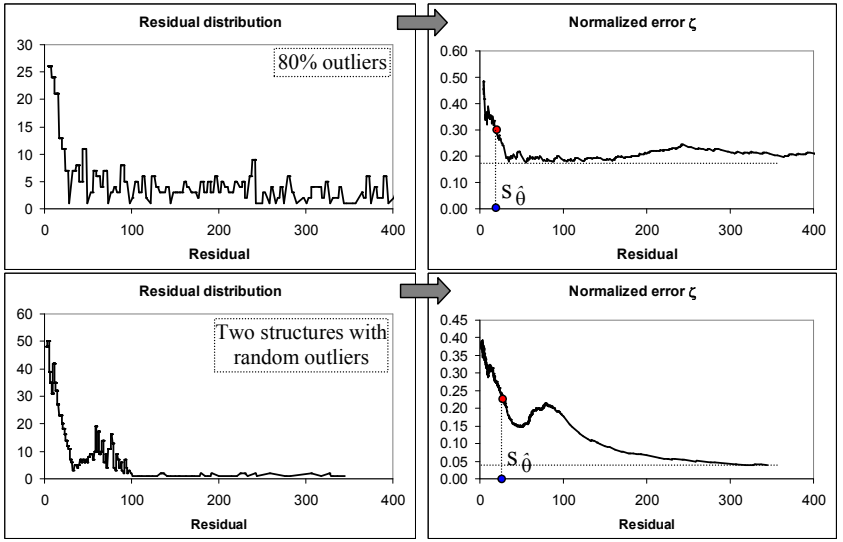

Fig. 7 Demonstration of finding scale estimate for adaptive bin-width computation through various situations of residual distribution.

(2) Find the min value of ζ_k:

$$\zeta_{min} = \min_{k_{max}<k\leq n}(\zeta_k).$$

(3) Find the first k_1, $k_1 > k_{max}$, that produces

$$\zeta_{k_1} = \frac{1}{2}(\zeta_{max} + \zeta_{min}).$$

(4) The scale estimate $s_{\hat{\theta}}$ for bin-width computation function (19) is set:

$$s_{\hat{\theta}} = r_{k_1}.$$

(5) Finaly the adaptive bin-width is computed as follows:

$$b_{\hat{\theta}} = \left(\frac{243\int_{-1}^{1}K(\zeta)^2 d\zeta}{35N(\int_{-1}^{1}\zeta^2 K(\zeta)d\zeta)^2}\right)^{\frac{1}{5}} r_{k_1}, \tag{23}$$

The demonstration of finding the scale estimate $s_{\hat{\theta}}$ for adaptive bin-width computation is described in **Fig. 7**:

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 inlier unimodal distribution with deviation σ of N residuals, the bin-width is computed, the densest bin at the mode contains a limited number of residuals, which is many greater than 1. Then, the number of bins for that inlier distribution is limited, 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.

6.4 Hypothesis Evaluation Function

Inspired by the use of the KDE in the pbM-Estimator[21)22)], and ASKC[13)], we also apply it in our adaptive objective function to evaluate the putative hypothesis:

$$F(\hat{\theta}) = \frac{1}{N\hat{\sigma}^*_{\hat{\theta}}} \sum_{i=1}^{N} K(\frac{r_{i,\hat{\theta}}}{\hat{\sigma}^*_{\hat{\theta}}}), \tag{24}$$

where $\hat{\sigma}^*_{\hat{\theta}}$ is adaptively estimated by the proposed inlier scale estimator as shown in Section 6 and κ has been defined in Section 6; K is a kernel such as Gaussian or the case-dependent kernel made in Section 5. 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.

7. Summary of Proposed Robust Estimators: FITSAC1, FITSAC2

In this paper, we introduce two robust estimators for the experiments: FITSAC1 and FITSAC2. The flowchart for these estimators is generalized as shown in Fig. 3. The difference between two estimators is described as follows. FITSAC2 is a completely adaptive estimator without any support from user. It uses the matching method in Section 6.2 and the binwidth in Section 6.3.2. FITSAC1 does not require the information about the noise on inliers, however it requires some user's support for computing the binwidth. It uses the matching method that is described in Section 6.1 and the residual density distribution is computed using the binwidth in Section 6.3.1.

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[27)]. In our experiments, we fix the same number of iterations for the proposed method as well as the compared methods.

8. Experiments with Adaptive-scale Robust Estimators

In this section, we describe the experiments carried out to validate our algorithms 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 experiments with real data are carried out to validate the algorithm in real situation. For the plane and line fitting problems, we compared our algorithms 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[35)]. 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 FITSAC1, 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. For FITSAC2, the inlier scale, or the standard deviation of inlier residuals, is estimated. The performance of FITSAC2 does not rely on any user-define parameters. However, when the exact inlier detection is necessary, we also use κ to distinguish inliers, but this is done after the execution of random sampling. The criteria for validating the proposed estimators 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.

8.1 Linear Residual

In this problem, a 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 estimators 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:

$$r_i = |\hat{a}x_i + \hat{b}y_i + \hat{c}z_i + \hat{d}|, \tag{25}$$

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

$$Error_{\hat{\theta}} = \sqrt{(a-\hat{a})^2 + (b-\hat{b})^2 + (c-\hat{c})^2 + (d-\hat{d})^2}, \tag{26}$$

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

8.1.1 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 10000 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.

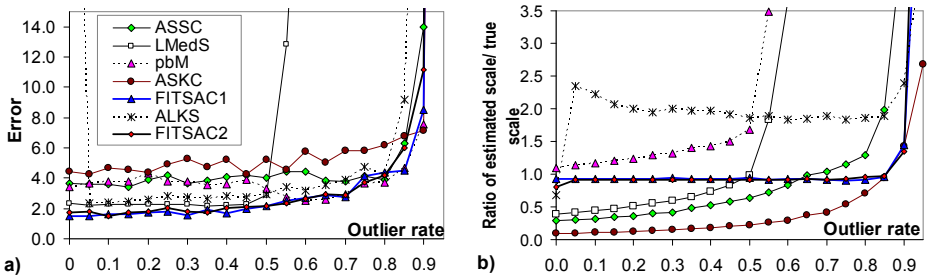

Fig. 8 Experiments with varying outlier rates for single-line data: (a) estimation error, (b) ratio between the scale of the estimated inlier residuals and the scale of residuals of true inliers.

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. 8**. Fig. 8.a describes the break-down point and the accuracy of the robust estimators, while Fig. 8.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 estimators for estimating residual density does not really depend on the bandwidth (or bin-width), and thus the accuracy of the pbM and proposed estimators remain high for the various outlier rates.

In addition, as the proposed estimators 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.

8.1.2 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. 9**, while the average results are shown in **Fig. 10**. Fig. 10.a describes the estimation error, while Fig. 10.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 estimators gives the most robust performance. FITSAC1 and FITSAC2 have quite similar performance. These results confirm that our proposed estimators have the best accuracy and robustness of all the estimators, and the estimated inlier bound is quite close to the ground-truth.

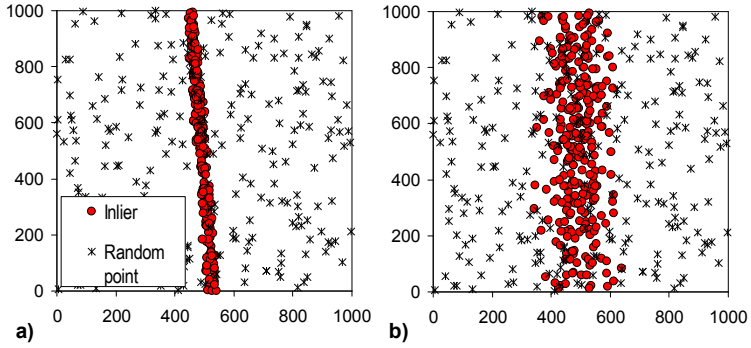

Fig. 9 Random data sets with an outlier rate of 60% and (a) $\sigma_G = 8.0$ and (b) $\sigma_G = 50$.

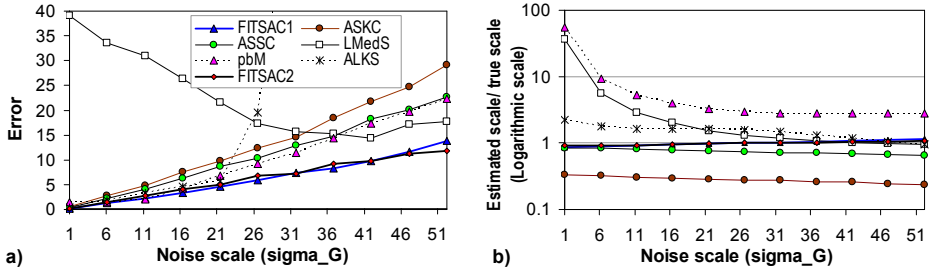

Fig. 10 Experiments with varying Gaussian noise scales and outlier rate fixed at 60%. Proposed estimators are highly resistant to high noise levels.

8.1.3 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:

$$\begin{aligned} Line1 : 2x - y + d &= 0, \ where\ d = 20, 30, 40, ...210 \\ Line2 : 2x - y &= 0. \end{aligned}$$

Various random data sets were used, with each data set containing 270 random outliers, 420 random points on $line2$, and 210 random points on $line1$. Gaussian noise σ_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. 11**, in which FITSAC1 and FITSAC2 have similar results. 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 inliers, while the proposed estimators estimated the inliers correctly. The average results for 100 random data sets are shown in **Fig. 12**. Fig. 12.a shows the estimation error for the robust estimators, while Fig. 12.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

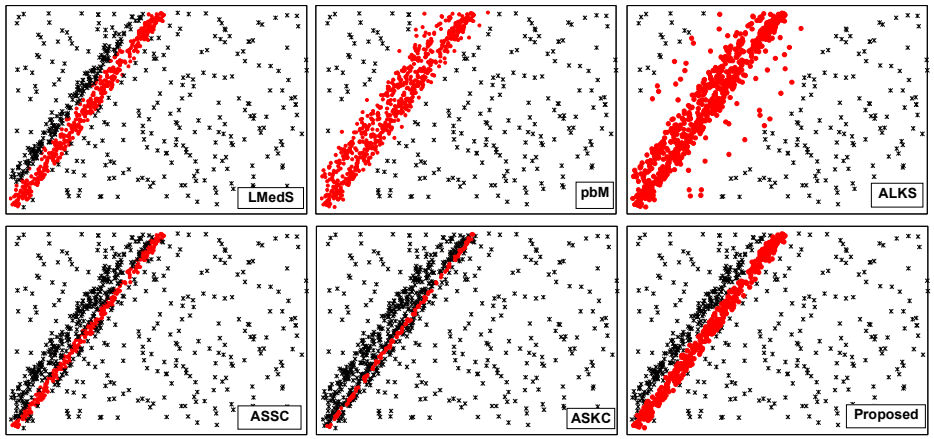

Fig. 11 Parallel Lines: estimation by each estimator using a random data set with d=70. ALKS and the pbM are confused, since the two lines are extracted as one. ASKC and ASSC extract a small part of the actual line. LMedS estimates a line with a large number of inliers belonging to *Line* 1 and a few inliers belonging to *Line* 2. The proposed methods (FITSAC1 and FITSAC2) extract one of the two lines correctly and neatly.

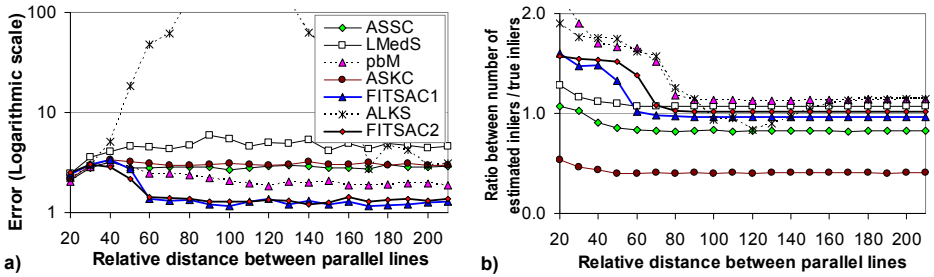

Fig. 12 Parallel Lines: (a) estimation error, (b) ratio between the number of estimated inliers and the number of true inliers.

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[8] 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 estimators give 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. FITSAC1 can detect the inliers better than FITSAC2 when two lines are close since it ignores the long tail of the distribution model, while FITSAC2 use the whole model for matching. However, Here, the results of FITSAC2 has shown that even-though it does not limit the distribution model for matching, FITSAC2 still resist to the multiple outlier distribution quite well. The reason for this resistant is that the structure with high density is always detected first in the proposed algorithm, outlier distribution from other structures has only small effect on the whole matching function and the ground distribution parameter compensates for the whole outlier distribution well.

8.1.4 Range Image Segmentation

In this experiment, we demonstrate the ability of robust estimators for segmentation problem, for example, range image segmentation. In this problem, a robust estimator must extract all the planes that make the object, a chair in our experiment. The segmentation for each robust estimator is done as follows. First, the robust estimator extracts one segment, then the inliers for that segment are removed from the data. The same procedure is repeated with a fixed number of iterations or until there is no remaining data points. We set this number to 8 for this chair image.

The segmentation results are shown in **Fig. 13**. The results show that pbM over-detected the inliers then some structures were combined. ASKC and ASSC under-detected the inliers then some planes were divided into smaller parts. FITSAC1 and FITSAC2 gave the most proper results, in which planes were segmented clearly.

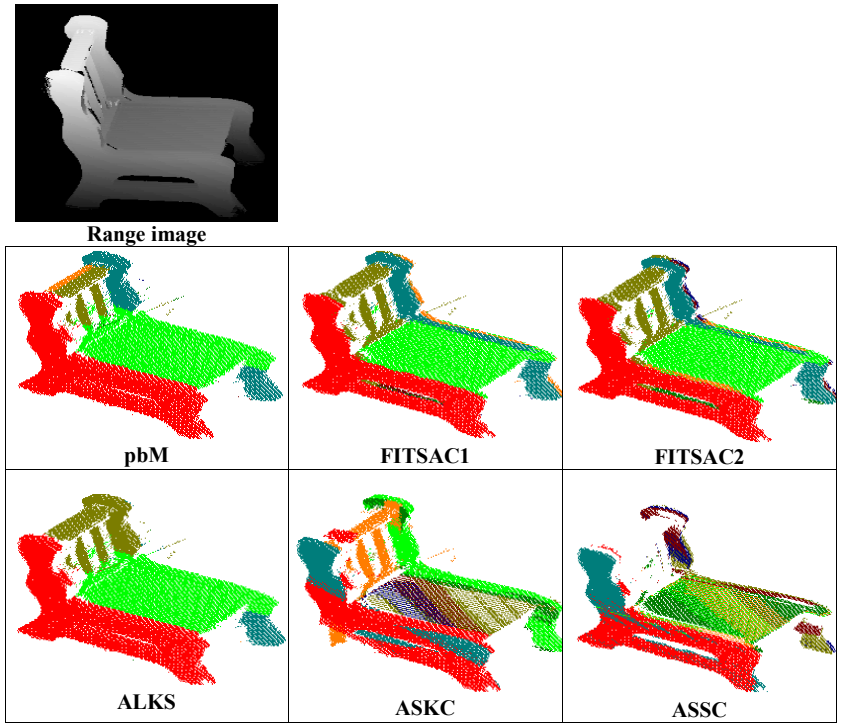

Fig. 13 Range image segmentation demonstrates the ability of a robust estimator for segmentation problem. The robust estimator must properly detect the inliers. The over-detection or under-detection results in a undesirable segmentation.

8.2 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 $\hat{\boldsymbol{\theta}}^* = \hat{\boldsymbol{F}}^*$:

$$Error_{\hat{\boldsymbol{F}}^*} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(r_{i,\hat{\boldsymbol{F}}^*})^2}, \tag{27}$$

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.

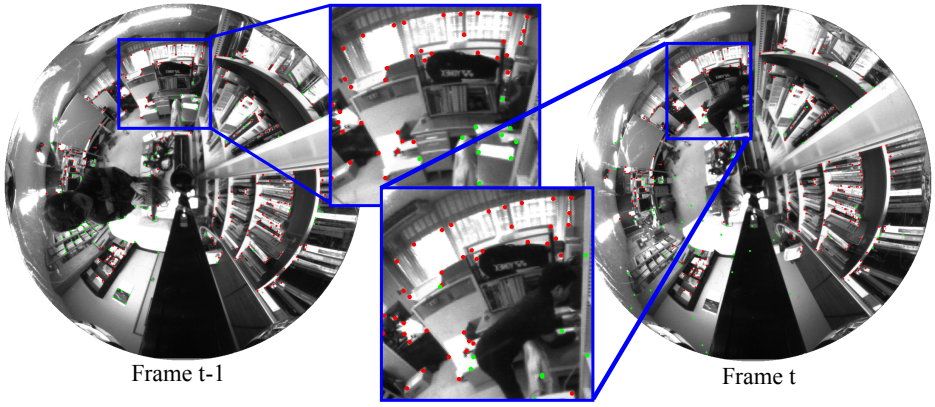

Fig. 14 A pair of images in a sequence: inliers (image features in red) and outliers (image features in green) are output by the proposed estimator.

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. 14**. 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[29] implemented in OpenCV[28]. 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 $Video\_4deg$, $Video\_14deg$ and $Video\_18deg$ for rotation speeds of 4 degrees/frame, 14 degrees/frame, and 18 degrees/frame, respectively. We computed the error by (27) 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 $Video\_4deg$ is low, about 10%. For $Video\_14deg$, the outlier rate is about 50%, and for $Video\_18deg$, 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 FITSAC1 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. FITSAC2 is slightly less accurate than FITSAC1 but the inlier detection is similar to that of FITSAC1. FITSAC2 has the similar accuracy compared to ASSC. 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.

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			
FITSAC1	0.000615	0.001493	0.001692
FITSAC2	0.000638	0.001690	0.001765
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
FITSAC1	182.307	110.847	87.935
FITSAC2	183.306	109.913	82.925
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

8.3 Computational Cost

We simulated the relation between processing time and the number of data points, the average results of which are shown in **Fig. 15**. For the FITSAC1, 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. FITSAC2 is the second slowest estimator. 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.

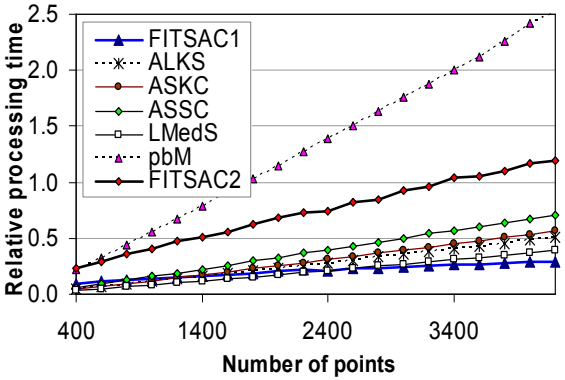

Fig. 15 Processing time for all estimators

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.

9. Conclusions

In this paper, we proposed two novel highly robust estimators (FITSAC1 and FITSAC2) for the estimation problem in computer vision that deals with data with high outlier rates and multiple structures. Our algorithms do not need any prior information about the inlier scale, as it 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 adaptive smoothing parameter efficiently help FITSAC2 work robustly in various situation without any support from user.

The proposed robust estimators were 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 estimators 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.

References

1) A. Hornberg, Handbook of Machine Vision, Wiley VCH, 2006.
2) A. J. Lacey, N. Pinitkarn, N. A. Thacker, An Evaluation of the Performance of RANSAC Algorithms for Stereo Camera Calibration, British Machine Vision Conference, 2008.
3) R.I. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2nd Edition, 2004.
4) T. Okabe, Y. Sato, Object Recognition Based on Photometric Alignment Using RANSAC, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 221–228, 2003.
5) P.J. Rousseeuw and A. Leroy. Robust Regression and Outlier Detection. John Wiley & Sons, New York. 1987.
6) 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.
7) 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.
8) K.M. Lee, P. Meer, and R.H. Park, Robust adaptive segmentation of range images, IEEE Trans. Pattern Anal. Machine Intelligence, Vol.20, pp. 200–205, 1998.
9) C. V. Stewart, MINPRAN: A new robust estimator for computer vision, IEEE Trans. Pattern Anal. Machine Intelligence, Vol.17, pp. 925–938, 1995.
10) X. Yu, T.D. Bui, A. Krzyzak, "Robust Estimation for Range Image Segmentation and Reconstruction," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 5, pp. 530-538, May, 1994.
11) H. Chen and P. Meer, Robust regression with projection based M-estimators, 9th Intl. Conf. on Computer Vision, pp. 878-E85, 2003.
12) H. Wang and D. Suter, Robust Adaptive-Scale Parametric Model Estimation for Computer Vision, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.26, No.11, pp. 1459–1474, 2004.
13) 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, Conference on Computer Vision and Pattern Recognition,

2008.
14) J. Choi; G. Medioni, StaRSaC: Stable Random Sample Consensus for Parameter Estimation, IEEE Conference on Computer Vision and Pattern Recognition, pp. 675–682, 2009.
15) J. Illingworth and J. Kittler, A survey of the Hough transform, Computer Vision, Graphics, and Image Processing (CVGIP), Vol.44, pp. 87–116, 1988.
16) L. Xu and E. Oja, Randomized Hough Transform (RHT): Basic mechanisms, algorithms, and computational complexities, CVGIP: Image Understanding, vol. 57, no. 2, pp. 131–154, 1993.
17) D.Nister, Preemptive RANSAC for live structure and motion estimation, Ninth IEEE International Conference on Computer Vision, pp. 199, 2003.
18) J. Matas and O. Chum, Randomized RANSAC with Sequential Probability Ratio Test, Proc. International Conference on Computer Vision, vol. 2, pp.1727–1732, 2005.
19) P.H.S. Torr and A. Zisserman, MLESAC: A new robust estimator with application to estimating image geometry, Computer Vision and Image Understanding, Vol. 78, pp. 138-E56, 2000.
20) P.H.S. Torr and D.W. Murray, Guided-MLESAC: Faster Image Transform Estimation by Using Matching Priors, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 27, No.10, pp.1523–1535, 2005.
21) S. Rozenfeld, I. Shimshoni, The Modified pbM-Estimator Method and a Runtime Analysis Technique for the RANSAC Family, Conference on Computer Vision and Pattern Recognition, pp. 1113–1120, 2005.
22) R. Subbarao, P. Meer, Beyond RANSAC: User Independent Robust Regression, p. 101, 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06), 2006.
23) P.H.S. Torr and D.W. Murray, The development and comparison of robust methods for estimating the fundamental matrix, Intl. J. Computer Vision, Vol.24, pp. 271–300, 1997.
24) H.C.Longuet-Higgins, A computer algorithm for reconstructing a scene from two projections, Nature 293, pp. 133–135, 1981.
25) 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.
26) 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.
27) D.A. Forsyth and J. Ponce, Computer Vision: A Modern Approach, Prentice Hall, 2002.
28) Intel Corporation, Open Source Computer Vision Library, http://www.intel.com/technology/computing/opencv/index.htm
29) J. Shi and C. Tomasi, Good Features to Track, Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 593–600, 1994.
30) M.P. Wand and M. Jones. Kernel Smoothing. Chapman & Hall. 1995.
31) V.C. Raykar, R. Duraiswami, Fast optimal bandwidth selection for kernel density estimation, In Proc. 2006 SIAM Conference on Data Mining, pp. 524–528, 2006.
32) G.R. Terrell, The Maximal Smoothing Principle in Density Estimation, Journal of the American Statistical Association, vol. 85, no. 410, pp. 470–477, 1990.
33) G.R. Terrell, D.W. Scott, Oversmoothed Nonparametric Density Estimates, Journal of the American Statistical Association, vol. 80, no. 389, pp. 209–214, 1985.
34) R.S Stephan , W.S David, On Locally Adaptive Density Estimation, Journal of the American Statistical Association, vol. 91, no. 436, pp. 1525–1534, 1996.
35) R. Subbarao, P. Meer, pbM-Estimator source code, http://www.caip.rutgers.edu/riul/research/robust.html