Robust Regression to Varying Data Distribution and Its Application to Landmark-based Localization

Sunglok Choi
U-Robot Research Division
Electronics and Telecommunications Research Institute
Daejeon, Republic of Korea
e-mail: sunglok@etri.re.kr

Jong-Hwan Kim
Department of Electrical Engineering and Computer Science
KAIST
Daejeon, Republic of Korea
e-mail: johkim@rit.kaist.ac.kr

***Abstract*—Data may be wrongly measured or come from other sources. Such data is a big problem in regression, which retrieve parameters from data. Random Sample Consensus (RANSAC) and Maximum Likelihood Estimation Sample Consensus (MLESAC) are representative researches, which focused on this problem. However, they do not cope with varying data distribution because they need to tune variables according to given data. This paper proposes user-independent parameter estimator, u-MLESAC, which is based on MLESAC. It estimates variables necessary in probabilistic error model through expectation maximization (EM). It also terminates adaptively using failure rate and error tolerance, which can control trade-off between accuracy and running time. Line fitting experiments showed its high accuracy and robustness in varying data distribution. Its results are compared with other estimators. Its application to landmark-based localization also verified its performance compared with other estimator.**

I. Introduction

Many engineering problems are to extract information from data. These include line fitting, conic fitting, camera calibration, and localization. Regression is mathematical generalization of these problems. Least squares method is a popular solution. However, it leads an incorrect result when some of data are wrongly measured or come from others sources. Such data are called as *outliers*, which lie outside the pattern of overall data. Other data with small noise are called as *inliers*.

M-estimator [1], least median of squares (LMedS) [2], Hough transform [3], and RANSAC [4] were proposed to overcome outliers in early statistic and computer vision. M-estimator and LMedS try to minimize other loss functions instead of sum of squared errors. For example, LMedS use median among squared errors. Hough transform finds the most frequent set of parameters in parameter space, so it needs huge amounts of memory to keep the parameter space. RANSAC is a sampling-based iterative algorithm. It estimates a preliminary set of parameters from randomly sampled data, then count the number of inlier candidates which have small error with respect to the estimated parameters. After repeating such procedure, it chooses the final parameters which have the maximal number of inlier candidates. RANSAC is popular until now due to its simple implementation. It have been a milestone of many further researches[1]. It also shares basic idea with recent researches for landmark-based localization [5], [6], [7]. Recent studies based on RANSAC can be categorized by their objectives – enhancing *accuracy*, reducing *running time*, and improving *robustness*. MLESAC [8] are representative works increasing accuracy of RANSAC. It is the first method to introduce a probabilistic error model and criterion. It chooses parameters which maximize likelihood of data. Many researchers used prior knowledge [10], [11] to replace random sampling to guided sampling. Guided sampling makes their estimator find desired parameters earlier. Randomized RANSAC (R-RANSAC) [12] used random sampling more thoroughly in contrast to them. It uses small amounts of data which are sampled randomly, when R-RANSAC counts inlier candidates, Most RANSAC-based approaches need to tune variables such as the number of iteration. The variables should be adjusted again when given data are changed. For example, when the ratio of outliers becomes bigger, more iteration is required. Self-tuning problem is investigated by projection-based M-estimator (pbM-estimator) [15], Feng and Hung's estimator [16], and AMLESAC [17]. pbM-estimator uses a nonparametric error model using kernel density estimation, which is contrast with the parametric model of MLESAC. Feng and Hung' estimator and AMLESAC adopt the error model from MLESAC. Both estimate variables of the error model using EM, gradient descent, and so on. Three studies are meaningful to achieve robustness against varying or unknown data. However, all do not consider the number of iteration deeply, which is a vital variable to sustain high accuracy in varying data situation.

This paper proposes a novel parameter estimator, u-MLESAC. It is based on MLESAC, so it is also repetition of four steps – sampling data, estimating parameters, estimating variables of the error model, and evaluating the parameters according to ML criterion. In contrast MLESAC, u-MLESAC estimates variance of the error model. It also calculates the proper number of iteration according to its terminal criterion. Its terminal criterion is to guarantee that two events happen simultaneously – all sampled data belong to inliers and have small noise enough to satisfy error tolerance. The number of iteration can be derived from probabilities of each event. Accuracy and running time can be adjusted by two conditions, failure rate and error tolerance.

[1]There was a birthday workshop for RANSAC, *25 Years of RANSAC*, in conjunction with CVPR 2006

1-4244-2384-2/08/$20.00 © 2008 IEEE

The remainder of this paper is organized as follows. Section II formulates nonlinear regression problem with outliers. Section III introduces u-MLESAC. It deals with the probabilistic error model and its estimation using EM. It also explains parameter evaluation via ML and adaptive termination using the probabilistic criterion. Section IV presents accuracy and running time of u-MLESAC compared with previous estimators. Line fitting problem are tackled in various data distribution. Section V demonstrates application to landmark-based localization. Finally, Section VI contains summary and further works.

II. Parameter Estimation Problem with Outlier

A. Problem Formulation

The true set of parameters $\tilde{M}$ is unknown. Data $\mathcal{D}$ are divided into inliers $\mathcal{D}_{in}$ and outliers $\mathcal{D}_{out}$, which satisfy the following conditions:

$$\mathcal{D} = \mathcal{D}_{in} \cup \mathcal{D}_{out} \text{ and } \mathcal{D}_{in} \cap \mathcal{D}_{out} = \emptyset. \tag{1}$$

Data class z_i represent whether a datum $\boldsymbol{d}_i$ is an inlier or outlier as the follows:

$$z_i = \begin{cases} 1, & \boldsymbol{d}_i \in \mathcal{D}_{in} \\ 0, & \boldsymbol{d}_i \in \mathcal{D}_{out} \end{cases}, \tag{2}$$

which is a hidden variable. Three functions are defined. A regression function is estimating parameters from a subset of data $\mathcal{S}$ as follows:

$$M = \text{Algo}(\mathcal{S}) \quad (\mathcal{S} \subset \mathcal{D}). \tag{3}$$

An error function is calculating error of a datum with respect to parameters M as follows:

$$e_i = \text{Err}(\boldsymbol{d}_i; M) \quad (e_i \in \mathbb{R}). \tag{4}$$

A loss function is evaluating risk of error as follows:

$$l_i = \text{Loss}(e_i) \quad (l_i \in \mathbb{R}). \tag{5}$$

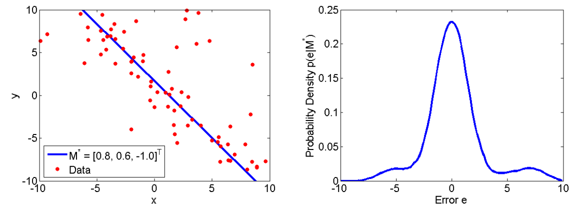
A general regression problem is formulated as

$$\hat{M} = \arg\min_{M} \left\{ \sum_{\boldsymbol{d} \in \mathcal{D}} \text{Loss}\Big(\text{Err}(\boldsymbol{d}; M)\Big) \right\}. \tag{6}$$

Least squares is a case when a loss function is square of error. This formulation regards whole data as inliers. If some of data are outliers, they can deteriorate the result. Therefore, the regression problem with outliers modified as

$$\hat{M} = \arg\min_{M} \left\{ \sum_{\boldsymbol{d} \in \mathcal{D}_{in}} \text{Loss}\Big(\text{Err}(\boldsymbol{d}; M)\Big) \right\}. \tag{7}$$

It is worse than a simple optimization problem, because the inlier set $\mathcal{D}_{in}$ is unknown.



(a) The true line and data (b) Error probability distribution

Fig. 1. An example: a line, data, and their error

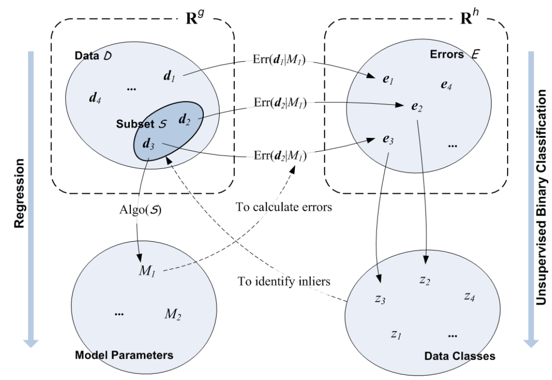


Fig. 2. Simultaneous Regression and Binary Data Classification

B. Simultaneous Regression and Binary Classification

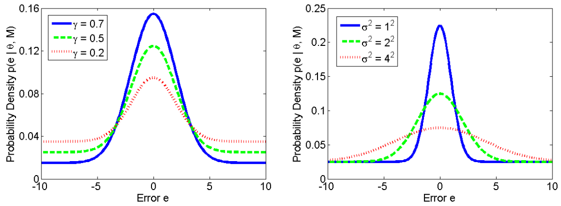
Error can be a clue to identify data class as data is a key to estimate parameters. For example, the true line and 100 data are given as Figure 1(a). Error is calculated through a signed Euclidian distance between the line and each point. Details are in Table I. Data will be different when the given line is changed. However, error distribution is seldom changed as Figure 1(b). Inliers and outliers can be determined by the magnitude of their error under the assumption that the true line is known. RANSAC uses a predefined threshold to decide whether a datum is an inlier or outlier. However, the assumption is impossible because it was a goal of the regression problem. The situation is described as Figure 2, which is *simultaneous regression and binary classification* problem. RANSAC and its family solve such twisted problem through repetition of sampling. They hope that sampled data are all inliers, which give nearly true parameters. They repeat such sampling until their dream is probable sufficiently. u-MLESAC also follow this approach.

III. u-MLESAC: Robust Parameter Estimator

A. Probabilistic Error Model

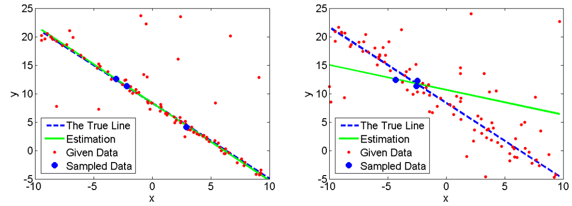
Error probability density function (pdf) $\text{p}(e|M)$ is expressed as

$$\begin{aligned} \text{p}(e|M) &= \text{P}(z=1|M)\text{p}(e|z=1,M) \\ &\quad + \text{P}(z=0|M)\text{p}(e|z=0,M), \end{aligned} \tag{8}$$



(a) $\sigma^2 = 2^2$ (b) $\gamma = 0.5$

Fig. 3. Torr and Zisserman' model ($\nu = 20$)



(a) $\sigma^2 = 0.25^2$ (b) $\sigma^2 = 1.0^2$

Fig. 4. Line fitting when sampled data are all inliers ($m = 3$)

where $\mathrm{P}(z = j|M)$ is prior probability and $\mathrm{p}(e|z = j, M)$ is error pdf of an inlier ($j = 1$) or outlier ($j = 0$). A posterior probability is derived as

$$\mathrm{P}(z = j|e, M) = \frac{\mathrm{P}(z = j|M)\mathrm{p}(e|z = j, M)}{\mathrm{p}(e|M)}. \tag{9}$$

u-MLESAC uses Torr and Zisserman [8] error pdf. It models inlier error pdf as *unbiased Gaussian distribution* and outlier error pdf as *uniform distribution* as follows:

$$\mathrm{p}(e|M) = \gamma \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{e^2}{2\sigma^2}\right) + (1-\gamma)\frac{1}{\nu}, \tag{10}$$

where γ is inlier prior probability $\mathrm{P}(z = 1|M)$ and ν is the size of error space. Therefore, its posterior probabilities become

$$\mathrm{P}(z = 0|e, M) = 1 - \mathrm{P}(z = 1|e, M) \quad \text{and} \tag{11}$$

$$\mathrm{P}(z = 1|e, M) = \frac{\gamma \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{e^2}{2\sigma^2}\right)}{\gamma \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{e^2}{2\sigma^2}\right) + (1-\gamma)\frac{1}{\nu}} \tag{12}$$

Torr and Zisserman' model is parameterized as two variables, γ and σ^2. The variable γ has physical meaning, which is ratio of inliers to whole data. The variable σ^2 means variance of Gaussian noise, that is, the magnitude of inlier noise. Figure 3 shows error pdf according to various values, which is similar with Figure 1(b).

B. Error PDF Estimation

u-MLESAC estimates γ and σ^2 using EM, because it is necessary to calculate probability density $\mathrm{p}(e = e_i|M)$, shortly $\mathrm{p}(e_i|M)$. EM is a popular algorithm for finding variables in probabilistic models, where the models have hidden variables. It is repetition of E-step and M-step. E-step calculates expectation of likelihood with respect to all possible cases of hidden variables. M-step finds variables of probabilistic models, which maximize the expectation of likelihood. In case of Torr and Zisserman's model, γ and σ^2 are estimated as follows:

$$\gamma = \frac{1}{n}\sum_{i=1}^{n} w_i \quad \text{and} \quad \sigma^2 = \frac{\sum_{i=1}^{n} w_i e_i^2}{\sum_{i=1}^{n} w_i}, \tag{13}$$

where w_i is posterior probability. $\mathrm{P}(z_i = 1|e_i, M)$ (12). Initial values of γ and σ^2 is

$$\gamma_{init} = 0.5 \quad \text{and} \quad \sigma_{init}^2 = \mathrm{median}(e_1^2, e_2^2, \cdots, e_n^2), \tag{14}$$

which are assigned before iterating two steps. Figure 5 contains its brief flow.

C. Parameter Evaluation

u-MLESAC uses ML criterion to select proper parameters among many sets of parameters, which come from each iteration. Likelihood is a measure how data $\mathcal{D}$ are probable with respect to given parameters M, which is noted as $\mathrm{p}(\mathcal{D}|M)$. ML criterion is to choose parameters which have the biggest likelihood, which means that data are the most feasible with the selected parameters. Error pdf $\mathrm{p}(e|M)$ is used instead of unknown data pdf $\mathrm{p}(\boldsymbol{d}|M)$. Under Naïve assumption, the likelihood becomes

$$\mathrm{p}(\mathcal{E}|M) = \prod_{i=1}^{n} \mathrm{p}(e_i|M). \tag{15}$$

This paper uses negative log likelihood as follows:

$$\mathrm{NLL}(M) = -\ln \prod_{i=1}^{n} \mathrm{p}(e_i|M) = -\sum_{i=1}^{n} \ln \mathrm{p}(e_i|M), \tag{16}$$

which make a small likelihood value numerically possible in digital computer. The problem is formulated as

$$\hat{M} = \arg\min_{M} \mathrm{NLL}(M). \tag{17}$$

D. Adaptive Termination

Adaptive termination is important because redundant iteration consumes unnecessary time and insufficient iteration does not guarantee proper estimates. Fischer and Bolles [4] proposed a probabilistic approach at first. They used probability to sample an inlier among the whole data, which is the same with the inlier ratio γ. If m data are sampled each iteration, it attempt to guarantee a condition – *sampled data are all inliers at least once among t trails with failure rate α*. It gives the number of iteration as follows:

$$t = \frac{\log \alpha}{\log(1-\gamma^m)}. \tag{18}$$

Feng and Hung [16] applied this calculation to their adaptive termination. However, it is not enough. Incorrect estimation is possible even if the condition is satisfied. Figure 4 shows that a set of inliers can give wrong estimation.

u-MLESAC calculates the necessary number of iteration using two conditions – 1) *sampled data are all iniliers* and 2) *they are within desired error tolerance β*. The necessary number of iteration is calculated as the follows:

$$t = \frac{\log \alpha}{\log(1 - k^m\gamma^m)} \quad \text{where} \quad k = \mathrm{erf}\left(\frac{\beta}{\sqrt{2}\sigma}\right), \tag{19}$$

CONFIGURATION VARIABLES

α : Failure rate (0.01 is used in this paper.)

β : Error tolerance

γ_{min} : Lower bound of γ (0.3 is used in this paper.)

δ_{em} : Tolerance of EM iteration (0.001 is used in this paper.)

PROCEDURE u-MLESAC

$t_{max} \leftarrow \log \alpha / \log(1 - \gamma_{min}^{m})$ $\quad \rightarrow$ Equation (18)

$t \leftarrow t_{max}$

$loss_{min} \leftarrow \infty$

$iteration \leftarrow 0$

WHILE UNTIL $iteration < t$

$\quad iteration \leftarrow iteration + 1$

1. Sample data randomly.

$\quad \mathcal{S} \leftarrow$ random samples of $\mathcal{D}$ $\quad (\mathrm{N}(\mathcal{D}) = n, \mathrm{N}(\mathcal{S}) = m)$

2. Estimate M from sampled data.

$\quad M \leftarrow \mathrm{Algo}(\mathcal{S})$

3. Calculate $\mathcal{E}$ with respect to M

$\quad \mathcal{E} \leftarrow \mathrm{Err}(\mathcal{D}; M)$

4. Estimate γ and σ^2 using EM.

$\quad \gamma \leftarrow 0.5$ $\quad \rightarrow$ Equation (14).

$\quad \sigma^2 \leftarrow \mathrm{median}(\mathcal{E})$ $\quad \rightarrow$ Equation (14).

$\quad$ DO

$\quad\quad \gamma_{prev} = \gamma$

$\quad\quad \gamma \leftarrow \frac{1}{n} \sum_{i=1}^{n} w_i$ $\quad \rightarrow$ Equation (13).

$\quad\quad \sigma^2 \leftarrow \frac{\sum_{i=1}^{n} w_i e_i^2}{\sum_{i=1}^{n} w_i}$ $\quad \rightarrow$ Equation (13).

$\quad$ WHILE UNTIL $|\gamma - \gamma_{prev}| < \delta_{em}$

5. Evaluate M using ML.

$\quad loss \leftarrow \mathrm{NLL}(M)$ $\quad \rightarrow$ Equation (16).

$\quad$ IF $loss < loss_{min}$ THEN

$\quad\quad loss_{min} \leftarrow loss$

$\quad\quad M_{best} \leftarrow M$

$\quad\quad t \leftarrow \frac{\log \alpha}{\log(1 - k^m \gamma^m)}$ $\quad \rightarrow$ Equation (19).

$\quad$ ENDIF

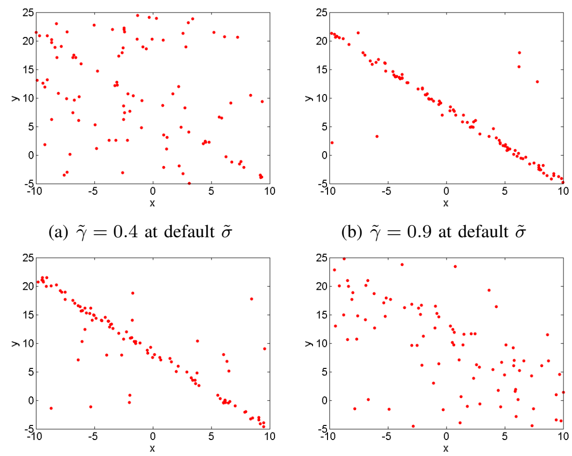
ENDWHILE

RETURN M_{best}

Fig. 5. Pseudo Code of u-MLESAC

Model	$ax + by + c = 0 \ (a^2 + b^2 = 1)$
Model Parameters	$M = [a, b, c]^T$
Datum	$\boldsymbol{d}_i = [x_i, y_i]^T$
Error Func.	$\mathrm{Err}(x_i, y_i; a, b, c) = ax_i + by_i + c$
Regression Func.	Least Squares ($m = 3$)
Loss Func.	$\mathrm{Loss}(e) = \lvert e \rvert$
The True Model	$\tilde{M} = [0.8, 0.6, -1.0]^T$
Data Space	$x_i \in [-10, +10], \ y_i \in [-5, +25]$
Error Space Size	$\nu = \sqrt{20^2 + 30^2}$
Default Condition	$(\tilde{\gamma}, \tilde{\sigma}) = (0.7, 0.25)$

TABLE I
LINE FITTING



(a) $\tilde{\gamma} = 0.4$ at default $\tilde{\sigma}$ (b) $\tilde{\gamma} = 0.9$ at default $\tilde{\sigma}$

(c) $\tilde{\sigma} = 0.25$ at default $\tilde{\gamma}$ (d) $\tilde{\sigma} = 3.00$ at default $\tilde{\gamma}$

Fig. 6. Line fitting in various data distribution

where erf is Gauss error function which is used to calculate a value of Gaussian cdf. Coefficient k has physical meaning, which is probability that sampled data belong to the error bound β. u-MLESAC can control trade-off between accuracy and running time using two variables α and β. Its overall procedure is described in Figure 5.

IV. Line Fitting Experiments

A. Configuration

Line fitting problem is used to measure accuracy and running time of u-MLESAC. 200 data are generated for each experiment. Inliers are generated as follows:

$$x_i = \tilde{x}_i + \delta x \text{ where } \delta x \sim \mathrm{N}(0, \tilde{\sigma}^2) \quad \text{and} \quad y_i = \tilde{y}_i + \delta y \text{ where } \delta y \sim \mathrm{N}(0, \tilde{\sigma}^2), \tag{20}$$

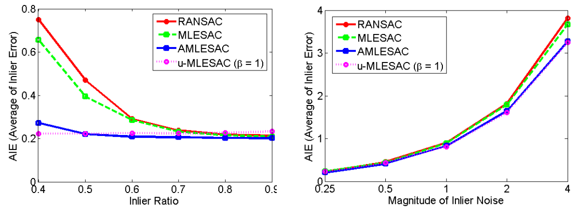
where $[\tilde{x}_i, \tilde{y}_i]^T$ is a point on the true line and $\tilde{\sigma}^2$ is variance of Gaussian noise. Outliers are generated randomly within given data space. Ratio of inliers to whole data is $\tilde{\gamma}$. Details are in Table I. Average of inlier error (AIE) is used as accuracy measure, which is

$$\mathrm{AIE}(M; \mathcal{D}_{in}) = \frac{1}{\mathrm{N}(\mathcal{D}_{in})} \sum_{\boldsymbol{d}_i \in \mathcal{D}_{in}} \Big| \mathrm{Err}(\boldsymbol{d}_i; M) \Big|. \tag{21}$$

AIE comes from the problem definition (7). Running time is measured by MATLAB `clock` function at Intel Core 2 CPU 2.13GHz. The experiment is performed in two sets of varying data – 1) *varying inlier ratio* $\tilde{\gamma}$ and 2) *varying magnitude of Gaussian noise* $\tilde{\sigma}^2$. Four representative situations is described in 6. 200 runs were evaluated on each condition for statistically meaningful results. RANSAC [4], MLESAC [8], and AMLESAC [17] are also performed for comparison. Their tuning variables are adjusted at default experiment condition in Table I.

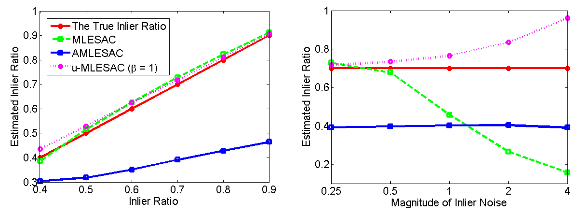
B. Results and Discussion

1) Accuracy: AMLESAC and u-MLESAC had small AIE regardless of varying inlier ratio (Figure 7(a)). However, AIE of RANSAC and MLESAC became increasing under 0.7 inlier ratio. The number of iteration used in RANSAC and MLESAC was adjusted at 0.7 inlier ratio, so they did not had enough number of iteration under 0.7 – it was worse condition than default. As the magnitude of noise became larger, AIE of four estimators also increased (Figure 7(b)). It resulted from definition of AIE, which is average of inlier 'noise'. However, AIE of AMLESAC and u-MLESAC was smaller than that of RANAC and MLESAC when the noise became larger.



(a) AIE in varying $\tilde{\gamma}$ (b) AIE in varying $\tilde{\sigma}$

Fig. 7. Line fitting: Accuracy (Average of Inlier Error)



(a) Estimated γ in varying $\tilde{\gamma}$ (b) Estimated γ in varying $\tilde{\sigma}$

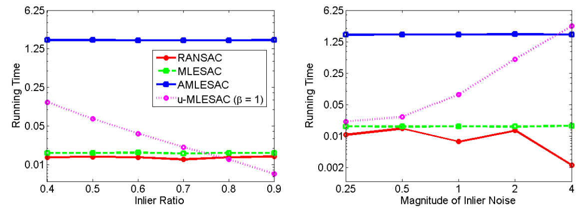
Fig. 8. Line fitting: Estimated inlier ratio γ

2) Error PDF Estimation: MLESAC and u-MLESAC estimated inlier ratio γ near the truth (Figure 8(a)). However, MLESAC did not cope in the case of varying magnitude of noise. MLESAC needs to tune variance of its error model, so it does not estimate inlier ratio beyond its tuned variance. AMLESAC had huge error between the truth and its estimation. RANSAC does not take into account of any model, so its estimation of inlier ratio is not in the Figure 8. Estimating the magnitude of noise σ^2 also similar results.

3) Running Time: The number of iteration in AMLESAC is determined by the worst situation. Therefore, it has accurate results in varying data distribution, but its running time is 100 times more than others (Figure 9). The number of iteration in RANSAC and MLESAC is tuned at default condition, so their running time also does not change in varying data situations. However, running time of u-MLESAC was varied according to each experiment situation. It had shorter running time than RANSAC and MLESAC under 0.7 inlier ratio (Figure 9(a)). Moreover, it runs longer when the magnitude of noise became larger. A situation with large noise is hard to estimate parameters accurately (see Figure 6(c) and 6(d)). Therefore, u-MLESAC repeated its procedure more to satisfy its error tolerance.

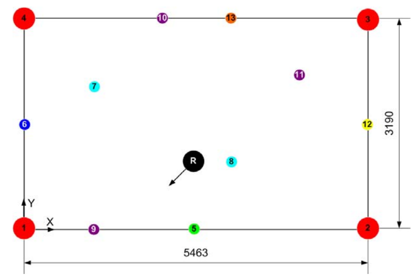
V. Application to Landmark-based Localization

Localization is one of the most important tasks for mobile robots to perform complex tasks such as guidance and security. Landmark-based localization is widely used due to its simplicity. It only needs position of landmarks to find location of a robot. However, it is troubled by outliers, which result from ambiguity of natural landmarks, imperfect landmark identification, dynamic obstacles, and so on. Se et al. [6] and Yuen and MacDonald [7] utilized RANSAC to overcome outliers.



(a) Running time in varying $\tilde{\gamma}$ (b) Running time in varying $\tilde{\sigma}$

Fig. 9. Line fitting: Running time



(a) Map: landmarks and a robot(R)

(b) Picture: landmarks and a robot

Fig. 10. An environment for landmark-based localization

Choi and Kim [18] used MLESAC. This paper introduces an application of u-MLESAC to landmark-based localization problem.

A. Configuration

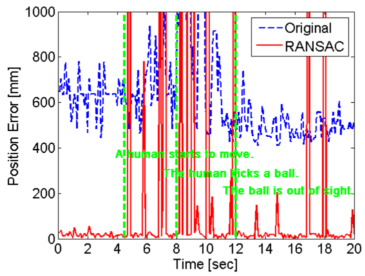
Given environment is presented in Figure 10. There were 13 landmarks. Some of them had identical size and color, which caused outliers. A robot was at $(2281, 1006)$ with orientation -140 degree. Table II contains details. Three events occurred during the experiment – 1) a human moved around a robot, 2) the human kicked a ball whose color was the same with a landmark, and 3) the ball went away from the robot. Original, RANSAC, and MLESAC were performed together in the same situation for comparison. Original was least squares method without any robust estimator.

B. Results and Discussion

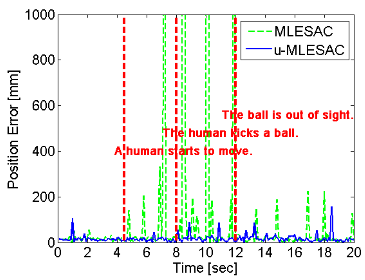
Localization with u-MLESAC was the most robust in ambiguity of landmarks and the complex environment (Figure 11(b)). Localization with RANSAC and MLESAC suffered huge position error when three events occurred. MLESAC

Model Parameters	$M = [x, y, \theta]^T$ (unit: [mm], [rad])
Datum	Bearing Angle: β_i Landmark Position in the Map: $[\tilde{x}_i, \tilde{y}_i]^T$
Error Func.	$\text{Err}(\beta_i; M) = \angle R(\angle\vec{d} - \theta)R(\beta_i)^T$ $(\vec{d} = [\tilde{x}_i, \tilde{y}_i]^T - [x, y]^T)$
Regression Func.	Least Squares Method using Bearings [19]
Loss Func.	$\text{Loss}(e) = \lvert e \rvert$
The True Model	$\tilde{M} = [2281, 1006, -2.44]^T$
Data Space	$\beta_i \in [-\pi, +\pi],\ \tilde{x}_i \in [0, 5463],\ \tilde{y}_i \in [0, 3190]$
Error Space Size	$\nu = 2\pi$

TABLE II
LANDMARK-BASED LOCALIZATION



(a) Position Error: Original and RANSAC



(b) Position Error: MLESAC and u-MLESAC

Fig. 11. Application to landmark-based localization

seems to better than RANSAC. Localization with only least square method (ORIGINAL) had big and constant position error all the time.

VI. CONCLUSION

u-MLESAC can attain high accuracy in varying data distribution. It does not have tuning variables, which need a tedious tuning task. Such task is substituted by error pdf estimation using EM and adaptive termination. Moreover, failure rate α and error tolerance β can control accuracy and running time, which are trade-off. Its performance was verified by experiments and an application to landmark-based localization.

u-MLESAC is a general framework to strengthen previous estimators, so it can be applied to other problems such as function fitting, camera calibration, image matching, outlier removal, and so on. It is a meaningful research to determine the proper number of samples, m, by u-MLESAC. R-RANSAC can be incorporated with u-MLESAC to accelerate estimating time.

ACKNOWLEDGMENT

The authors would like to thank Taemin Kim for his sincere discussion and comments.

REFERENCES

[1] P. J. Huber, *Robust Statistics*. John Wiley and Sons, 1981.
[2] P. Rousseeuw, "Least median of squares regression," *Journal of the American Statistical Association*, vol. 79, no. 388, pp. 871–880, 1984.
[3] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, pp. 11–15, January 1972.
[4] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, June 1981.
[5] L. Jaulin, M. Kieffer, E. Walter, and D. Meizel, "Guaranteed robust nonlinear estimation with application to robot localization," *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Application and Reviews*, vol. 32, no. 4, pp. 374–381, November 2002.
[6] S. Se, D. G. Lowe, and J. J. Little, "Vision-based global localization and mapping for mobile robots," *IEEE Transaction on Robotics*, vol. 21, no. 3, pp. 364–375, June 2004.
[7] D. C. K. Yuen and B. A. MacDonald, "Vision-based localization algorithm based on landmark matching, triangulation, reconstruction, and comparison," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 217–226, April 2005.
[8] P. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, pp. 138–156, 2000.
[9] O. Chum, J. Matas, and S. Obdrzalek, "Enhancing RANSAC by generalized model optimization," in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2004.
[10] B. Tordoff and D. W. Murray, "Guided sampling and consensus for motion estimation," in *Proceedings of the 7th European Conference on Computer Vision (ECCV)*, vol. 1. Springer-Verlag, 2002, pp. 1470–1477.
[11] O. Chum and J. Matas, "Matching with PROSAC - Progressive Sample Consensus," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.
[12] J. Matas and O. Chum, "Randomized RANSAC with sequential probability ratio test," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV)*, 2005.
[13] D. Myatt, P. Torr, S. Nasuto, J. Bishop, and R. Craddock, "NAPSAC: High noise, high dimensional robust estimation - it's in the bag," in *Preceedings of the 13th British Machine Vision Conference (BMVC)*, 2002, pp. 458–467.
[14] V. Rodehorst and O. Hellwich, "Genetic Algorithm SAmple Consensus (GASAC) - a parallel strategy for robust parameter estimation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2006.
[15] H. Chen and P. Meer, "Robust regression with projection based M-estimator," in *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV)*, vol. 2, October 2003, pp. 878–885.
[16] C. Feng and Y. Hung, "A robust method for estimating the fundamental matrix," in *Proceedings of the 7th Digital Image Computing: Techniques and Applications*, no. 633–642, December 2003.
[17] A. Konouchine, V. Gaganov, and V. Veznevets, "AMLESAC: A new maximum likelihood robust estimator," in *Proceedings of the International Conference on Computer Graphics and Vision (GrapiCon)*, 2005.
[18] S. Choi and J.-H. Kim, "Reducing effect of outliers in landmark-based spatial localization using MLESAC," in *Proceedings of the IFAC World Congress*, July 2008.
[19] I. Shimshoni, "On mobile robot localization from landmark bearings," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 6, pp. 971–976, December 2002.