Performance Evaluation of RANSAC Family

Sunglok Choi[1]
sunglok@etri.re.kr
Taemin Kim[2]
taemin.kim@nasa.gov
Wonpil Yu[1]
ywp@etri.re.kr

[1] Robot Research Department
ETRI
Daejeon, Republic of Korea

[2] Intelligent Robotics Group
NASA Ames Research Center
Moffett Field, CA

Abstract

RANSAC (Random Sample Consensus) has been popular in regression problem with samples contaminated with outliers. It has been a milestone of many researches on robust estimators, but there are a few survey and performance analysis on them. This paper categorizes them on their objectives: being accurate, being fast, and being robust. Performance evaluation performed on line fitting with various data distribution. Planar homography estimation was utilized to present performance in real data.

1 Introduction

The various approaches to robust estimation of a model from data have been studied. In practice, the difficulty comes from outliers, which are observed out of pattern by the rest of data. Random Sample Consensus (RANSAC) [11] is widely applied to such problems due to its simple implementation and robustness. It is now common context in many computer vision textbooks, and there was also its birthday workshop, *25 Years of RANSAC*, in conjunction with CVPR 2006.

There were the meaningful groundwork before RANSAC. M-estimator, L-estimator, R-estimator [15] and Least Median of Squares (LMedS) [24] were proposed in the statistics field. They formulate regression with outliers as a minimization problem. This formulation is similar with least square method, which minimize sum of squared error values. However, they use nonlinear and rather complex loss functions instead of square of error. For example, LMedS tries to minimize median of error values. They need a numerical optimization algorithm to solve such nonlinear minimization problem. Hough transform [9] was also suggested in the image processing field. It transforms data (*e.g.* 2D points from a line) from data space into parameter space (*e.g.* slop and intercept). It selects the most frequent point in the parameter space as its estimation, so it needs huge amounts of memory to keep the parameter space.

RANSAC simply iterates two steps: generating a hypothesis from random samples and verifying it to the data. It does not need complex optimization algorithms and huge amounts of memory. These advantages originate from random sampling, which is examined in Section 2. It is applied to many problems: model fitting, epipolar geometry estimation, motion estimation/segmentation, structure from motion mobile, and feature-based localization.

© 2009. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.
BMVC 2009 doi:10.5244/C.23.81

Numerous methods have been derived from RANSAC and form their family. There has been a few and old survey and comparison on them [19, 29, 31]. An insightful view of the RANSAC family is also necessary. They can be categorized by their research objectives: being *accurate*, being *fast*, and being *robust* (Figure 2). Representative works are briefly explained in Section 3. Their performance is evaluated through line fitting and planar homography estimation in Section 4. Concluding remarks and further research direction are made in Section 5.

2 RANSAC Revisited

RANSAC is an iterative algorithm of two phases: *hypothesis generation* and *hypothesis evaluation* (Figure 1).

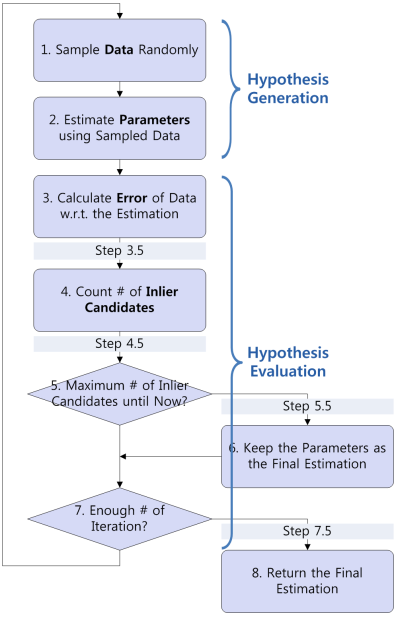


Figure 1: Flowchart of RANSAC

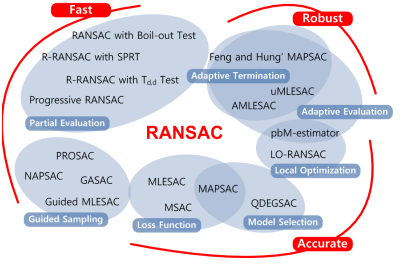


Figure 2: RANSAC Family

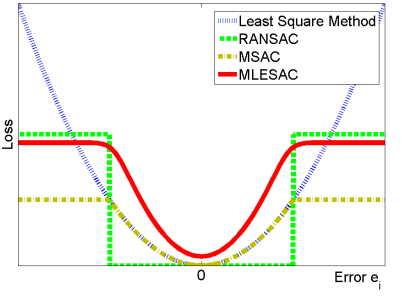


Figure 3: Loss Functions

2.1 Hypothesis Generation

RANSAC picks up a subset of data randomly (Step 1), and estimates a parameter from the sample (Step 2). If the given model is line, $ax+by+c=0$ $(a^2+b^2=1)$, $M=[a,b,c]^T$ is the parameter to be estimated, which is a hypothesis of the truth. RANSAC generates a number of hypotheses during its iteration.

RANSAC is not regression technique such as least square method and Support Vector Machine. It uses them to generate a hypothesis. It wraps and strengthens them, which do not sustain high accuracy if some of data are outliers. RANSAC uses a portion of data, not whole data. If the selected data are all inliers, they can entail a hypothesis close to the truth. This assumption leads the necessary number of iteration enough to pick up all inlier samples

at least once with failure probability α as follows:

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

where m is the number of data to generate a hypothesis, and γ is probability to pick up an inlier, that is, ratio of inliers to whole data (shortly *the inlier ratio*). However, the inlier ratio γ is unknown in many practical situations, which is necessary to be determined by users.

RANSAC converts a estimation problem in the continuous domain into a selection problem in the discrete domain. For example, there are 200 points to find a line and least square method uses 2 points. There are ${}_{200}C_2 = 19,900$ available pairs. The problem is now to select the most suitable pair among huge number of pairs.

2.2 Hypothesis Evaluation

RANSAC finally chooses the most probable hypothesis, which is supported by the most inlier candidates (Step 5 and 6). A datum is recognized as the inlier candidate, whose error from a hypothesis is within a predefined threshold (Step 4). In case of line fitting, error can be geometric distance from the datum to the estimated line. The threshold is the second tuning variable, which is highly related with magnitude of noise which contaminates inliers (shortly *the magnitude of noise*). However, the magnitude of noise is also unknown in almost all application.

RANSAC solves the selection problem as an optimization problem. It is formulated as

$$\hat{M} = \arg\min_{M} \left\{ \sum_{d \in \mathscr{D}} \mathrm{Loss}\Big(\mathrm{Err}(d;M)\Big) \right\}, \tag{2}$$

where $\mathscr{D}$ is data, Loss is a loss function, and Err is a error function such as geometric distance. The loss function of least square method is represented as $\mathrm{Loss}(e) = e^2$. In contrast, RANSAC uses

$$\mathrm{Loss}(e) = \begin{cases} 0 & |\,e\,| < c \\ \text{const} & \text{otherwise} \end{cases}, \tag{3}$$

where c is the threshold. Figure 3 shows difference of two loss functions. RANSAC has constant loss at large error while least square method has huge loss. Outliers disturb least squares because they usually have large error.

3 RANSAC's Descendants

3.1 To Be Accurate

Loss Function A loss function is used in evaluating a hypothesis (Step 4), which select the most probable hypothesis. MSAC (M-estimator SAC) [30] adopts bounded loss of RANSAC as follows:

$$\mathrm{Loss}(e) = \begin{cases} e^2 & |\,e\,| < c \\ c^2 & \text{otherwise} \end{cases}. \tag{4}$$

MLESAC (Maximum Likelihood SAC) [30] utilizes probability distribution of error by inlier and outlier to evaluate the hypothesis. It models inlier error as unbiased Gaussian distribution

and outlier error as uniform distribution as follows:

$$\mathrm{p}(e|M) = \gamma \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{e^2}{2\sigma^2}\right) + (1-\gamma)\frac{1}{v}, \tag{5}$$

where v is the size of available error space. γ is prior probability of being an inlier, which means the inlier ratio in (1). σ is the standard deviation of Gaussian noise, which is common measure of the magnitude of noise. MLESAC chooses the hypothesis, which maximize likelihood of the given data. It uses negative log likelihood for computational feasibility, so the loss function becomes

$$\mathrm{Loss}(e) = -\ln \mathrm{p}(e|M). \tag{6}$$

Figure 3 shows typical loss function of RANSAC, MSAC, and MLESAC, which are quite similar. RANSAC does not consider the amount of error within bounded threshold, so it has worse accuracy than other two methods. MAPSAC (Maximum A Posterior Estimation SAC) [28] follows Bayesian approach in contrast to MLESAC, but its loss function is similar with MSAC and MLESAC since it assumes uniform prior. pbM-estimator (Projection-based M-estimator) [25] attempted non-parametric approach in contrast to the parametric error model (5).

Local Optimization Local optimization can be attached at Step 5.5 or 7.5 to improve accuracy. LO-RANSAC (Locally Optimized RANSAC) [8] adds local optimization at Step 5.5. Chum reported that inner RANSAC with iteration was the most accurate among his proposed optimization schemes. He also proved that computation burden is increased $\log N$ times than RANSAC without it, where N is the number of data. pbM-estimator adopts complex but powerful optimization scheme at Step 5.5. To use pbM-estimator, the given problem should be transformed into EIV (Error-in-Variables) form as follows:

$$\mathbf{y}_i^T \theta - \alpha = 0 \quad (i = 1, 2, \cdots, N). \tag{7}$$

In case of line fitting, $\mathbf{y}_i$ is $[x_i, y_i]^T$, θ is $[a, b]^T$, and α is $-c$. The given problem becomes an optimization problem as follows:

$$\left[\hat{\theta}, \hat{\alpha}\right] = \arg\min_{\theta,\alpha} \frac{1}{N} \sum_{i=1}^{N} \kappa\left(\frac{\mathbf{y}_i^T \theta - \alpha}{s}\right), \tag{8}$$

where κ is M-kernel function and s is bandwidth. Two methods are utilized to find θ and α: conjugate gradient descent for θ and mean-shift for α. The optimization scheme can find θ and α accurately although the initial estimation is slightly incorrect.

Model Selection Model selection can be incorporated with RANSAC, which makes reasonable estimation even if data are degenerate for the given model. Torr proposed GRIC (Geometric Robust Information Criterion) [28] as fitness measure of models. It takes account of model and structure complexity. GRIC presented higher accuracy than AIC, BIC, and MDL, whose experiment was incorporated with MAPSAC. QDEGSAC (RANSAC for Quasi-degenerate Data) [12] has model selection step after Step 8. The model selection step has another RANSAC for models which need less parameters than the given model. It does not require domain-specific knowledge, but it uses the number of inlier candidates for checking degeneracy.

3.2 To Be Fast

Computing time of RANSAC is

$$T = t(T_G + NT_E), \tag{9}$$

where T_G is time for generating a hypothesis from sampled data, T_E is time for evaluating the hypothesis for each datum.

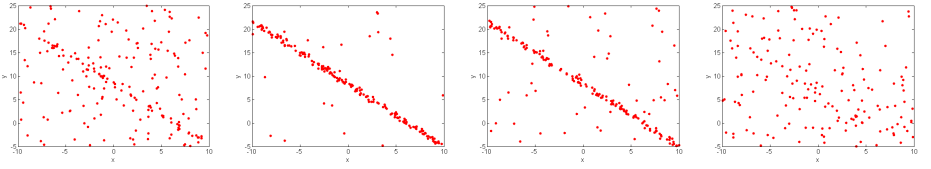
Guided Sampling Guided sampling can substitute random sampling (Step 1) to accelerate RANSAC. Guidance reduces the necessary number of iteration, t in (9), than random sampling. However, it can make RANSAC more slower due to additional computation burden, and it can impair potential of global search. Guided MLESAC [27] uses available cues to assign prior probabilities of each datum in (5). Tentative matching score can be used as the cues in estimating planar homography. PROSAC (Progressive SAC) [6] uses prior knowledge more deeply. It sorts data using the prior knowledge such as matching score. A hypothesis is generated among the top-ranked data, not whole data. PROSAC progressively tries less ranked data, whose extreme case is whole data. In contrast to Guided MLESAC and PROSAC, NAPSAC (N Adjacent Points SAC) [20] and GASAC (Genetic Algorithm SAC [23] do not use prior information. NAPSAC uses heuristics that an inlier tends to be closer to other inliers than outliers. It samples data within defined radius from a randomly selected point. GASAC is based on genetic algorithm, whose procedure is quite different from RANSAC (Figure 1). It manages a subset of data as a gene, which can generate a hypothesis. Each gene receives penalty by loss of its generated hypothesis. The penalty causes less opportunity in evolving the new gene pool from the current genes.

Partial Evaluation It is possible to quit evaluating a hypothesis if the hypothesis is far from the truth apparently, which reduces computing time. Partial evaluation decreases the necessary number of data for evaluation, N in (9). R-RANSAC (Randomized RANSAC) [5] performs a preliminary test, $T_{d,d}$ test, at Step 3.5. Full evaluation is only performed when the generated hypothesis passes the preliminary test. $T_{d,d}$ test is passed if all d data among randomly selected d data are consistent with the hypothesis. Another R-RANSAC [18] utilizes SPRT (Sequential Probability Ratio Test), which replaces Step 4 and 5. SPRT terminates when likelihood ratio falls under the optimal threshold, whose iterative solution was derived in [18]. Bail-out test [3] also similarly substitutes Step 4 and 5. Preemptive RANSAC [21] performs parallel evaluation, while RANSAC do serial evaluation. It generates enough number of hypotheses before its evaluation phase, and evaluates the hypotheses all together at each datum. It is useful for applications with limited time constraint.

3.3 To Be Robust

RANSAC needs to tune two variables with respect to the given data: the threshold c for evaluating a hypothesis and the number of iteration t for generating enough hypotheses.

Adaptive Evaluation The threshold c needs to be adjusted automatically to sustain high accuracy in in varying data. LMedS does not need any tuning variable because it tries to minimize median squared error. However, it becomes worse where the inlier ratio is under 0.5 since the median is from outliers. MLESAC and its descendants are based on the parametric



(a) $(\gamma^*, \sigma^*) = (0.3, 0.25)$ (b) $(\gamma^*, \sigma^*) = (0.9, 0.25)$ (c) $(\gamma^*, \sigma^*) = (0.7, 0.25)$ (d) $(\gamma^*, \sigma^*) = (0.7, 4.00)$

Figure 4: Examples of Line Fitting Data

probability distribution (5), which needs σ and γ for evaluating a hypothesis. Feng and Hung' MAPSAC [10] (not original MAPSAC [28]) and uMLESAC [4] calculate them through EM (Expectation Maximization) at Step 3.5. AMLESAC [16] uses uniform search and gradient descent for σ and EM for γ. pbM-estimator also needs bandwidth s (8) for its optimization step. It is determined as $N^{-0.2} \times MAD$, where MAD is Median Absolute Deviation.

Adaptive Termination The number of iteration t means the number of generated hypotheses, whose automatic assignment achieves high accuracy and less computing time in varying data. Feng and Hung' MAPSAC updates the number of iteration via 1 at Step 5.5, where γ comes from EM. However, it can terminate too earlier because incorrect hypotheses sometimes entail high γ with big σ (refer ambiguity of Figure 4(a) and 4(d)). uMLESAC updates the number via γ and σ together as follows:

$$t = \frac{\log \alpha}{\log(1 - k^m \gamma^m)} \quad \text{where} \quad k = \operatorname{erf}\left(\frac{\beta}{\sqrt{2}\sigma}\right), \tag{10}$$

where erf is Gauss error function which is used to calculate a value of Gaussian cdf. The variable k physically means probability that sampled data belong to the error confidence β. uMLESAC controls trade-off between accuracy and computing time using α and β. R-RANSAC with SPRT also has a similar discounting coefficient with k, which is probability of rejecting a good hypothesis in SPRT.

4 Experiment

4.1 Line Fitting: Synthetic Data

4.1.1 Configuration

Line fitting was tackled to evaluate performance of RANSAC family. 200 points were generated. Inliers came from the true line, $0.8x + 0.6y - 1.0 = 0$, with unbiased Gaussian noise. Outliers were uniformly distributed in the given region (Figure 4). Experiments were performed on various sets of the inlier ratio γ^* and magnitude of noise σ^* (Figure 4). Each configuration was repeated 1000 times for statistically representative results. Least square method using 3 points was used to estimate a line and geometric distance was used as an error function. Experiments were carried out on 12 robust estimators. Each estimator was tuned at each configuration except RANSAC* and MLESAC*, which were tuned only at $(\gamma^*, \sigma^*) = (0.7, 0.25)$ to compare robustness without tuning. Performance of each estimator was measured by accuracy and computing time. Accuracy was quantified through *the*

Inlier Ratio	0.300	0.400	0.500	0.600	0.700	0.800	0.900
LMedS	**27.320**	**6.461**	1.321	1.356	1.408	1.379	1.509
RANSAC	1.305	1.323	1.326	1.330	1.390	1.415	1.483
MSAC	1.229	1.266	1.284	1.337	1.373	1.415	1.535
MLESAC	1.248	1.269	1.289	1.316	1.358	1.410	1.446
LO-RANSAC	1.245	1.229	1.229	1.221	1.229	1.253	1.255
R-RANSAC.T	1.317	1.323	1.304	1.341	1.394	1.401	1.475
R-RANSAC.S	**15.210**	1.848	1.131	1.229	1.291	1.370	1.389
FH' MAPSAC	**7.708**	**2.036**	1.647	1.484	1.463	1.490	1.526
AMLESAC	**2.051**	1.477	1.452	1.529	1.517	1.526	1.575
GASAC	**28.640**	**7.370**	1.108	1.077	1.100	1.120	1.147
pbM-estimator	1.023	1.034	1.209	1.255	1.286	1.291	1.355
uMLESAC	**5.246**	1.382	1.402	1.383	1.398	1.433	1.489
RANSAC*	**47.010**	**13.920**	**3.031**	1.688	1.386	1.255	1.187
MLESAC*	**50.110**	**10.240**	**2.839**	1.694	1.352	1.235	1.145

Mag. of Noise	0.250	0.500	1.000	2.000	4.000
LMedS	1.379	1.372	1.295	1.256	1.228
RANSAC	1.385	1.380	1.315	1.282	1.259
MSAC	1.389	1.319	1.306	1.246	1.193
MLESAC	1.364	1.341	1.309	1.262	1.218
LO-RANSAC	1.219	1.224	1.202	1.203	1.222
R-RANSAC.T	1.396	1.363	1.286	1.232	1.180
R-RANSAC.S	1.253	1.325	1.413	1.583	1.618
FH'MAPSAC	1.458	1.557	1.513	1.395	1.406
AMLESAC	1.479	1.499	1.942	1.192	1.164
GASAC	1.098	1.100	1.102	1.115	1.124
pbM-estimator	1.258	1.272	1.264	1.296	1.306
uMLESAC	1.361	1.232	1.069	1.025	1.051
RANSAC*	1.389	1.379	1.409	1.440	1.552
MLESAC*	1.357	1.326	1.330	1.388	1.468

Figure 5: Accuracy (NSE) on Varying γ^* and σ^*

Inlier Ratio	0.300	0.400	0.500	0.600	0.700	0.800	0.900
LMedS	0.226	0.094	0.047	0.025	0.015	0.009	0.005
RANSAC	0.220	0.091	0.046	0.025	0.015	0.009	0.005
MSAC	0.216	0.089	0.045	0.024	0.014	0.009	0.005
MLESAC	0.257	0.107	0.053	0.029	0.017	0.011	0.006
LO-RANSAC	0.232	0.098	0.051	0.029	0.018	0.012	0.008
R-RANSAC.T	0.220	0.091	0.045	0.011	0.008	0.006	0.005
R-RANSAC.S	0.038	0.052	0.055	0.012	0.011	0.017	0.029
FH'MAPSAC	0.046	0.052	0.038	0.024	0.015	0.010	0.006
AMLESAC	**1.336**	0.520	0.248	0.129	0.072	0.044	0.025
GASAC	**1.366**	0.569	0.286	0.157	0.093	0.060	0.036
pbM-estimator	**3.390**	**1.438**	0.731	0.406	0.236	0.155	0.091
uMLESAC	**1.072**	0.139	0.068	0.036	0.021	0.012	0.007
RANSAC*	0.014	0.014	0.015	0.014	0.014	0.014	0.015
MLESAC*	0.017	0.017	0.017	0.017	0.017	0.017	0.017

Mag. of Noise	0.250	0.500	1.000	2.000	4.000
LMedS	0.015	0.015	0.015	0.015	0.015
RANSAC	0.014	0.014	0.014	0.014	0.014
MSAC	0.014	0.014	0.014	0.014	0.014
MLESAC	0.017	0.017	0.017	0.017	0.017
LO-RANSAC	0.018	0.018	0.018	0.018	0.018
R-RANSAC.T	0.008	0.009	0.011	0.013	0.020
R-RANSAC.S	0.011	0.013	0.044	0.036	0.011
FH'MAPSAC	0.015	0.014	0.012	0.029	**2.379**
AMLESAC	0.072	0.072	0.070	0.072	0.078
GASAC	0.093	0.093	0.093	0.093	0.093
pbM-estimator	**0.239**	**0.244**	**0.243**	**0.221**	**0.208**
uMLESAC	0.021	0.027	0.084	0.486	**2.763**
RANSAC*	0.014	0.014	0.014	0.014	0.014
MLESAC*	0.017	0.017	0.017	0.017	0.017

Figure 6: Computing on Varying γ^* and σ^*

normalized squared error of inliers (NSE),

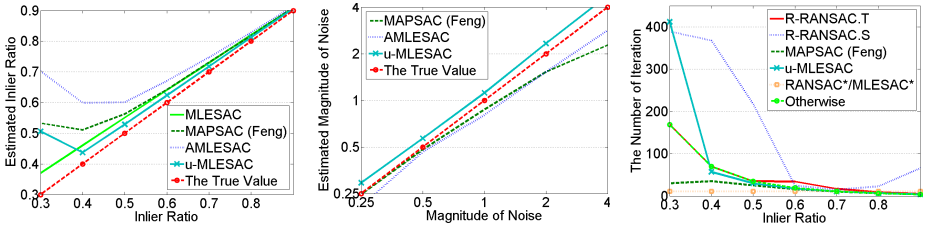
$$\mathrm{NSE}(M) = \frac{\sum_{d_i \in \mathscr{D}_{in}} \mathrm{Err}(d_i; M)^2}{\sum_{d_i \in \mathscr{D}_{in}} \mathrm{Err}(d_i; M^*)^2}, \tag{11}$$

where M^* and M are the true line and its estimation, $\mathscr{D}_{in}$ is a set of inliers. NSE comes from Choi and Kim' problem definition [4]. NSE is close to 1 when the magnitude of error by the estimated line is near the magnitude of error by the truth. Computing time was measured using MATLAB `clock` function at Intel Core 2 CPU 2.13GHz. Robustness can be observed via variation of accuracy in varying configuration.

4.1.2 Results and Discussion

Figure 5 and 6 show accuracy and computing time on the varying inlier ratio and magnitude of noise.

On Accuracy MLESAC were about 5% more accurate than RANSAC in almost all configurations. MLESAC takes into account the magnitude of error, while RANSAC has constant loss regardless of the magnitude. It makes better evaluation on hypotheses, but it became 15% slower than RANSAC. LO-RANSAC is about 10% more accurate than RANSAC due to its local optimization. It had about 5% more computation burden compared with RANSAC.



(a) Estimated γ on Varying γ^* (b) Estimated σ on Varying σ^* (c) Estimated t on Varying γ^*

Figure 7: Results on Varying Configuration

Similarly, GASAC and pbM-estimator were nearly 14% and 7% more accurate but 6% and 16 times slower than RANSAC.

On Computing Time R-RANSAC with $T_{d,d}$ test had similar accuracy with RANSAC, but slightly faster in spite of more iteration. It could reduce time because it used a subset of data in its evaluation step. R-RANSAC with SPRT had more accurate than RANSAC, but it did not reduce computing time due to its adaptive termination. Its adaptive scheme decided its termination much longer than tunned RANSAC (Figure 7(c)).

On Robustness MLESAC, Feng and Hung' MAPSAC, AMLESAC, u-MLESAC are based on an error model, which is mixture of Gaussian and uniform distribution. Figure 7(a) and 7(b) show estimated variables, γ and σ, of the error model, and Figure 7 presents the number of iteration. Feng and Hung' MAPSAC, AMLESAC and u-MLESAC estimated the error model approximately except low inlier ratio. Therefore, they had low accuracy near 0.3 inlier ratio, but they were more accurate than RANSAC* and MLESAC*, which were tunned at 0.7 inlier ratio, in low inlier ratio. LMedS and GASAC became less accurate under 0.5 inlier ratio. It results from their loss function, median of squared error, which selects squared error by an outlier when outliers is more than half. Especially, Feng and Hung' MAPSAC and u-MLESAC became much faster than RANSAC* and MLESAC* after 0.7 inlier ratio due to their adaptive termination. Their number of iteration at 0.3 inlier ratio were significantly different from the tunned number of iteration using 1 because of their incorrect error model estimation. R-RANSAC with SPRT overestimated the number of iteration in almost all configuration.

4.2 2D Homography Estimation: Real Data

4.2.1 Configuration

Planar homography estimation was also used to evaluate performance of the estimators. Oxford VGG *Graffiti* image set (Figure 8) were utilized for experiments. SIFT [17] were used to generate tentative corresponding points. Four-point algorithm [13] estimated planar homography using four sets of matched points. Accuracy and time measures were same with line fitting experiments. Error function Err is Euclidian distance between a point and projection of its matched point. Inliers were identified among tentative matching through the true homography, which is incorporated with *Graffiti* image set. pbM-estimator did not used to the experiments since it is difficult to transform homography constraint into EIV problem form.

(a) Image 1 (b) Image 2 (c) Image 3 (d) Image 4

Figure 8: Oxford VGG *Graffiti* Images [2]

Homography	1→2	1→3	1→4	Homography	1→2	1→3	1→4
LMedS	7.389	7.722	-	LMedS	0.600	2.096	2.843
RANSAC	5.648	8.345	6.631	RANSAC	0.601	2.092	2.832
MSAC	6.165	7.038	6.278	MSAC	0.598	2.082	2.891
MLESAC	5.955	8.633	6.705	MLESAC	0.626	2.163	2.947
LO-RANSAC	4.438	8.325	4.671	LO-RANSAC	0.707	2.188	2.904
R-RANSAC.T	6.609	7.897	6.088	R-RANSAC.T	0.178	0.309	0.362
R-RANSAC.S	3.678	-	-	R-RANSAC.S	0.198	1.016	1.615
FH' MAPSAC	6.294	-	-	FH' MAPSAC	0.484	5.180	5.509
AMLESAC	**13.460**	10.820	-	AMLESAC	1.436	4.066	4.685
GASAC	1.938	5.532	-	GASAC	3.713	**12.630**	**17.010**
u-MLESAC	5.067	8.460	2.995	u-MLESAC	0.780	3.280	6.400
RANSAC*	3.237	8.345	-	RANSAC*	3.942	2.092	1.202
MLESAC*	2.259	8.633	-	MLESAC*	4.092	2.163	1.225

Figure 9: Accuracy (NSE) and Computing time on Three Image Pairs

Experiments were performed into each image pair 100 times for statistically meaningful results.

4.2.2 Results and Discussion

Figure 9 shows accuracy and computing time on three image pairs, which are Image 1 to Image 2 (1640 pairs of points), Image 1 to Image 3 (888 pairs of points), and Image 1 to Image 4 (426 pairs of points). Dash (-) on Figure 9 means that its NSE is more than 50, that is, failure on the average. RANSAC* and MLESAC* were tunned in estimating homography from Image 1 to Image 3.

Accuracy of RANSAC, MSAC and MLESAC differed nearly 4%. MSAC was the most accurate among three estimators, and MLESAC was the worst. Local optimization in LO-RANSAC enhanced accuracy nearly 20% with 7% computational burden. GASAC also improved accuracy significantly, but its computing time was 5 times more than RANSAC.

Subset evaluation accelerated R-RANSAC with $T_{d,d}$ test around 6 times compared with RANSAC, which also kept similar accuracy with that. It is more significant improvement than line fitting experiments. R-RANSAC with SPRT also had similar performance in estimating homography from Image 1 to Image 2. However, it failed in other two experiments, which is similar result with 0.3 inlier ratio in line fitting experiments.

Performance of Feng and Hung' MAPSAC was similar with line fitting experiments. Its accuracy was similar with RANSAC, but it failed in severe situation as like homography from Image 1 to Image 4. Although u-MLESAC did not fail in the situation, but it also failed more severe image pairs which tunned RANSAC could estimate homography properly. They consumed less time in homography from Image 1 to Image 2 compared with RANSAC* and MLESAC*. LMedS and GASAC failed in estimating homography from Image 1 to Image 4, since its inlier ratio is less than 0.5.

5 Conclusion

RANSAC and its descendants are summarized in three viewpoints: accuracy, computing time, and robustness. This paper also describes that completely different methods share the same idea. For example, R-RANSAC and Preemptive RANSAC accelerate RANSAC though partial evaluation. Such insight is useful to analyze the previous works and develop the new method. The results of two experiments were also analyzed in three viewpoints. The results presented a *trade-off of accuracy/robustness and computing time*. For example, uMLESAC sustains high accuracy in varying data, but it needs 1.5 times more time than RANSAC due to its adaptation.

Meaningful researches has been performed in RANSAC family, but it needs to investigated more. Balanced accuracy/robustness and computing time can be achieved from merging the previous works or the new breakthrough. Adaptation in varying data is a challenging problem because the previous works do not keep accuracy in low inlier ratio. MLESAC is the first breakthrough which reformulated original RANSAC in the probabilistic view. The new interpretation of the problem can lead another breakthrough. The problem can be incorporated with another problems such as model selection. Data with multiple models are a attractive problem for the current single result formulation. The new tool can stimulate this field such as genetic algorithm of GASAC. Survey and performance evaluation including the recent works [7, 14, 26] contributes users to choose a proper method for their applications. The Middlebury Computer Vision Pages [1] is a good example for that.

Acknowlegement

The authors would like to thank Oxford VGG for their *Graffiti* images. This work was supported partly by the R&D program of the Korea Ministry of Knowledge and Economy (MKE) and the Korea Evaluation Institute of Industrial Technology (KEIT). (2008-S-031-01, Hybrid u-Robot Service System Technology Development for Ubiquitous City)

References

[1] Middlebury Computer Vision Pages. URL http://vision.middlebury.edu/.

[2] Oxford Visual Geometry Group Affine Covariant Regions Datasets. URL http://www.robots.ox.ac.uk/~vgg/data/data-aff.html.

[3] David Capel. An effective bail-out test for RANSAC consensus scoring. In *Proceedings of the British Machine Vision Conference (BMVC)*, September 2005.

[4] Sunglok Choi and Jong-Hwan Kim. Robust regression to varying data distribution and its application to landmark-based localization. In *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, October 2008.

[5] O. Chum and J Matas. Randomized RANSAC with $T_{d,d}$ test. In *Preceedings of the 13th British Machine Vision Conference (BMVC)*, pages 448–457, 2002.

[6] Ondrej Chum and Jiri Matas. Matching with PROSAC - Progressive Sample Consensus. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[7] Ondrej Chum and Jiri Matas. Optimal randomized ransac. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1472–1482, August 2008.

[8] Ondrej Chum, Jiri Matas, and Stepan Obdrzalek. Enhancing RANSAC by generalized model optimization. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2004.

[9] R. O. Duda and P. E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15:11–15, January 1972.

[10] C.L. Feng and Y.S. Hung. A robust method for estimating the fundamental matrix. In *Proceedings of the 7th Digital Image Computing: Techniques and Applications*, number 633–642, December 2003.

[11] 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*, 24(6):381–395, June 1981.

[12] Jan-Micheal Frahm and Marc Pollefeys. RANSAC for (quasi-)degenerate data (QDEGSAC). In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[13] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Combridge, 2 edition, 2003.

[14] Richard J. M. Den Hollander and Alan Hanjalic. A combined ransac-hough transform algorithm for fundamental matrix estimation. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2007.

[15] P. J. Huber. *Robust Statistics*. John Wiley and Sons, 1981.

[16] Anton Konouchine, Victor Gaganov, and Vladimir Veznevets. AMLESAC: A new maximum likelihood robust estimator. In *Proceedings of the International Conference on Computer Graphics and Vision (GrapiCon)*, 2005.

[17] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[18] 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.

[19] Peter Meer, Doran Mintz, Azriel Rosenfeld, and Dong Yoon Kim. Robust regression methods for computer vision: A review. *International Journal of Computer Vision*, 6 (1):59–70, 1991.

[20] D.R. Myatt, P.H.S Torr, S.J. Nasuto, J.M. 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)*, pages 458–467, 2002.

[21] David Nister. Preemptive RANSAC for live structure and motion estimation. In *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV)*, pages 199–206, 2003.

[22] Rahul Raguram, Jan-Michael Frahm, and Marc Pollefeys. A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008.

[23] Volker Rodehorst and Olaf 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.

[24] P.J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880, 1984.

[25] R. Subbarao and P. Meer. Subspace estimation using projection-based M-estimator over grassmann manifolds. In *Proceedings of the 9th European Conference on Computer Vision (ECCV)*, pages 301–312, May 2006.

[26] Raghav Subbarao and Peter Meer. Beyond RANSAC: User independent robust regression. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2006.

[27] Ben J. Tordoff and David W. Murray. Guided-mlesac: Faster image transform estimation by using matching priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1523–1535, October 2005.

[28] P.H.S. Torr. Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *International Journal of Computer Vision*, 50(1):35–61, 2002.

[29] P.H.S Torr and D.W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, 24(3): 271–300, 1997.

[30] P.H.S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:138–156, 2000.

[31] Zhengyou Zhang. Parameter estimation technique: A tutorial with application to conic fitting. *Image and Vision Computing*, 15(1):59–76, 1997.