



ISSN: 0976-3031

Available Online at <http://www.recentscientific.com>

CODEN: IJRSFP (USA)

*International Journal of Recent Scientific Research*  
Vol. 9, Issue, 12(A), pp. 29842-29846, December, 2018

**International Journal of  
Recent Scientific  
Research**

DOI: 10.24327/IJRSR

## Research Article

# EXTENSIONS TO THE STANDARD RANSAC ALGORITHM FOR EFFICIENCY AND ROBUSTNESS

**Wenyan Ci., Jue Wang., Mingxiang Zhu and Aiyu Dou**

School of Electric Power Engineering, Nanjing Normal University Taizhou College,  
Taizhou, China

DOI: <http://dx.doi.org/10.24327/ijrsr.2018.0912.2942>

### ARTICLE INFO

#### Article History:

Received 6<sup>th</sup> September, 2018  
Received in revised form 15<sup>th</sup>  
October, 2018  
Accepted 12<sup>th</sup> October, 2018  
Published online 28<sup>th</sup> December, 2018

#### Key Words:

RANSAC, Extensions, Robustness,  
Efficiency.

### ABSTRACT

Over the past decade, many improvements have been made to RANSAC, each of which addresses the specific weaknesses of the original algorithm. However, there are relatively few comprehensive studies on these developments. The purpose of this paper is to fill this gap by researching related technologies, so as to promote the development of new algorithms. This paper first introduces the standard RANSAC algorithm, and discusses its mechanism and limitations. Then, some extensions of RANSAC algorithm are introduced in detail. These extensions solve the limitations of robustness and efficiency. By combining these ideas, the performance of the algorithm can be further improved in terms of real-time performance and robustness.

**Copyright © Wenyan Ci et al, 2018**, this is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.

## INTRODUCTION

The RANSAC algorithm proposed by Fischler and Bolles (Fischler and Bolles, 1987) is a well-known method for estimating model parameters in the presence of outliers. It is based on iteration and hypothesis verification framework, and has robustness to contaminated observation data. The basic idea of RANSAC algorithm is to randomly extract a sample set from the complete set, calculate the model parameters, and then verify the other data points in the complete set with the obtained model parameters. After many iterations, the model parameters that can obtain the highest consistency in the data points are regarded as the solution of the model, while those data points which are inconsistent with the model parameters are regarded as the outliers. The size of the sample set is usually set to the minimum value that can be used to solve the model parameters. For example, the value is usually 3 in stereo vision. The standard RANSAC framework consists of hypothesis and verification stage, which are iterated over a set of observations. In the first stage, a set of randomly selected points is used to instantiate the model, in which the cardinality of the sampling set is equal to the minimum number of points needed to determine the parameters of the model, so the sampling set is also called the minimum set. The second stage of the algorithm is to determine the observation values

consistent with the model, which constitute a consistent set. The consistency here is usually defined based on some predefined error thresholds. After several iterations, the algorithm returns to the model that obtains the maximum consistent set.

RANSAC algorithm is an uncertain algorithm, i.e., the results of each execution of the algorithm may be different. It obtains reasonable results with a certain probability. The more iterations, the greater the probability. In fact, RANSAC algorithm has become a general method of excluding outliers in visual navigation system (Maimone *et al.*, 2007, Deigmoeller and Eggert, 2016). In recent years, some improved versions of RANSAC algorithm have appeared. Proposed methods which focus on consensus measure, sampling strategy, hypotheses verification are discussed in the paper.

### Consensus Measure

The loss function of RANSAC algorithm sets the value of inliers to 0, without considering the difference of error between inliers. When the setting of  $\tau^2$  is larger, more solutions of the same  $C$  will be generated, which makes the estimation of the model worse. This section introduces two robust consistency measurement methods.

\*Corresponding author: **Wenyan Ci**

School of Electric Power Engineering, Nanjing Normal University Taizhou College, Taizhou, China

**MSAC**

Inspired by M-estimator, Torr and Zisserman (Torr and Zisserman, 2000) proposed a new form of loss function:

$$\rho_{MAC}(e_i^2) = \begin{cases} e_i^2 & e_i^2 < \tau^2 \\ \tau^2 & e_i^2 \geq \tau^2 \end{cases} \quad (1)$$

At this point, the function value of the outliers is still a constant, but the function value of the inliers become related to the quality of data points and model fitting. This new estimator is called MSAC (M-estimator Sample Consensus). Literature (Torr and Zisserman, 2000) shows that the accuracy of MSAC is 5-10% higher than that of standard RANSAC for epipolar geometry problems.

**MLESAC**

MLESAC (Maximum Likelihood Estimation Sample Consensus) algorithm (Torr and Zisserman, 2000) has further developed the idea of MSAC. Instead, maximize the following objective functions:

$$C_{MLE} = \sum_i \log p(e_i | z) \quad (2)$$

where  $Z$  is the model parameter,  $\log p(e_i | z)$  represents the probability distribution model of error. In order to consider both noise and outliers, the error distribution of data points is expressed as a mixture model of Gauss distribution and uniform distribution. Assuming that the error of the inliers conform to the Gauss distribution  $N(0, \sigma)$  with standard deviation of  $\sigma$  and the outliers conform to the uniform distribution  $N(0, \sigma)$ , the following distribution model is established:

$$p_{MLE}(e_i | z) = \gamma \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{e_i^2}{2\sigma^2}\right) + (1-\gamma) \frac{1}{\nu} \quad (3)$$

where the mixed parameter  $\gamma$  is the proportion of the inliers in the observation set, and its value can be estimated by EM (Expectation Maximization) algorithm;  $\nu$  is a constant, reflecting the size of the search window in the matching algorithm. Similar to MSAC, MLESAC can improve the accuracy by 5-10% compared with the standard RANSAC algorithm, but the computational complexity is increased.

**Sampling Strategy**

In many implementations, it may be possible to incorporate prior information that allow observations to be scored according to how likely they are of being an inlier. This can have a dramatic effect on the efficiency of RANSAC, particularly for low inlier ratios. Several approaches take advantage of this fact to generate better sample sets.

**Guided-MLESAC**

One disadvantage of MLESAC is the need to calculate the mixed parameter  $\gamma$ . However, this parameter reflects the prior attributes of data and does not depend on a single model. It is better to assume that there is a larger  $C_{MLE}$  for any mixed parameter (Tordoff and Murray, 2002). Therefore, it is not required to solve the mixed parameter in each iteration. In

order to select the best hypothesis, it is sufficient to use a fixed mixing parameter (e.g.  $\gamma = 0.5$ ) for all hypotheses.

Furthermore, if we can get an independent priori value of whether the matching pair is correct or not, we can get better results than using global mixed parameters. Formula (3) can be rewritten as follows:

$$p_{G-MLE}(e_i | z) = p(v_i) \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{e_i^2}{2\sigma^2}\right) + (1-p(v_i)) \frac{1}{\nu} \quad (4)$$

where  $v_i$  is the marker variable of whether the data point  $i$  is an inlier, and  $p(v_i)$  is a prior probability.

In two view geometry, the prior probability can be obtained by matching scores of feature matching. Assuming that there are  $n_i$  possible matches of feature point  $i$ , their correctness is expressed by  $v_{ij}$ , and the matching score is expressed by  $s_{ij}$ , where  $j = 1 \dots n_i$ . In small baseline image pairing, the

matching score  $s_{ij}$  of mismatching  $v_{ij}$  can be empirically expressed as follows:

$$p(s_{ij} | v_{ij}) = \frac{3}{4} (1 - s_{ij})^2 \quad (5)$$

For correct matching, we have:

$$p(s_{ij} | v_{ij}) = a \frac{1 - s_{ij}}{\alpha^2} \exp\left[-\left(\frac{1 - s_{ij}}{\alpha}\right)^2\right] \quad (6)$$

where  $a$  is a normalized constant and  $\alpha$  is a "compactness" parameter. Suppose that all possible matches of the feature  $i$  have the same priori value, i.e.,  $p(v_{ij}) = 1 / (n_i + 1)$ ,  $p(v_{ij}) = n_i / (n_i + 1)$ . Finally, given all matching scores, the correct probability of matching  $ij$  can be obtained:

$$p(v_{ij} | s_{i,1 \dots n_i}) = \frac{p(v_{ij} | s_{ij}) \prod_{k \neq j} p(v_{ik} | s_{ik})}{\prod_{k=1}^{n_i} p(v_{ik} | s_{ik}) + \sum_{j=1}^{n_i} [p(v_{ij} | s_{ij}) \prod_{k \neq j} p(v_{ik} | s_{ik})]} \quad (7)$$

$p(v_{ij} | s_{i,1 \dots n_i})$  can be computed by  $p(s_{ij} | v_{ij})$  and  $p(s_{ij} | v_{ij})$ , and  $p(v_{ij} | s_{i,1 \dots n_i})$  can be used as an estimate of  $p(v_i)$  in formula  $p_{G-MLE}(e_i | z)$ .

In this way, the Monte-Carlo sampling method based on prior probability can be used for hypotheses generation. The higher the prior probability, the higher the probability of sampling points. This method can reduce the number of MLESAC iterations and improve the efficiency of the algorithm.

**PROSAC**

The PROSAC (Progressive Sample Consensus) algorithm (Chum and Matas, 2005) uses the quality of data points as a metric in order to priority generation the hypotheses that are more likely to be effective. Unlike RANSAC which extracts samples from all data, its samples are extracted from a subset of the highest quality datas. In fact, the samples extracted by PROSAC are the same as those extracted by RANSAC, but the sampling order is different. PROSAC begins with testing the most promising hypothesis. As the quality score of data points decreases, its sampling strategy gradually tends to RANSAC.

In PROSAC algorithm, matching pairs are arranged in descending order according to similarity. If the number of matching pairs is  $N$  and the corresponding ordered set is  $\Theta_N$ , then:

$$i < j \Rightarrow q(u_i) > q(u_j) \quad \forall u_i, u_j \in \Theta_N \quad (8)$$

where  $q(\cdot)$  is a similarity function.

Consider randomly extracting  $T_N$  samples from all  $N$  matching pairs, and the sequence is expressed as  $\{M_i\}_{i=1}^{T_N}$ .

Suppose that the sequence  $\{M_{(i)}\}_{i=1}^{T_N}$  contains the same samples as  $\{M_i\}_{i=1}^{T_N}$ , but it is arranged in order of sample quality from high to low. A subset of the complete set of  $\Theta_N$  is represented by  $\Theta_n$ , which contains  $n$  highest quality data points. Assuming that the sequence  $\{M_i\}_{i=1}^{T_n}$  contains  $T_n$  data points on average that are all from  $\Theta_n$ , then:

$$T_n = T_N \frac{\binom{n}{m}}{\binom{N}{m}} = T_N \prod_{i=0}^{m-1} \frac{n-i}{N-i} \quad (9)$$

Thus:

$$\frac{T_{n+1}}{T_n} = \frac{T_N \prod_{i=0}^{m-1} \frac{n+1-i}{N-i}}{T_N \prod_{i=0}^{m-1} \frac{n-i}{N-i}} = \frac{n+1}{n+1-m} \quad (10)$$

Finally, the recursive formula of  $T_{n+1}$  is obtained:

$$T_{n+1} = \frac{n+1}{n+1-m} T_n \quad (11)$$

$T_n$  samples come from  $\Theta_n$ , and  $T_{n+1}$  samples come from  $\Theta_{n+1}$ . Since subsets  $\Theta_{n+1}$  and  $\Theta_n$  satisfy  $\Theta_{n+1} = \Theta_n \cup u_{n+1}$ , there are  $T_{n+1} - T_n$  samples containing one data point  $u_{n+1}$  and  $m-1$  data points from  $\Theta_n$ . In order to generate  $M_{(i)}$  arranged in descending order according to sample quality,  $T_{n+1} - T_n$  samples can be extracted. These samples consist of

a data point  $u_{n+1}$  and  $m-1$  data points randomly sampled from  $\Theta_n$ , where  $n = m \dots N$ .

Because the good hypothesis is generated early in the sampling process, the efficiency of PROSAC is significantly improved compared with that of RANSAC. However, when quality scores are not very useful (e.g., in scenarios with significant repetitive structures), the improvements are not significant.

**Group SAC**

GroupSAC (Kai *et al.*, 2009) assumes that there are some natural groupings in the datas, some of which have a higher proportion of inliers, while others have a higher proportion of outliers. This kind of natural grouping exists in most of the problems we are interested in, e.g., feature pairs can be grouped according to optical flow in wide baseline matching.

In the standard RANSAC algorithm and many of its variants, it is assumed that the probability of one data point being an inlier is independent of other data points. For any minimal set  $S$  with  $m$  data points, the number of its inliers  $I_S$  complies with the distribution  $I_S \sim B(m, \varepsilon)$ . Among them,  $B(m, \varepsilon)$  represents binomial distribution;  $\varepsilon$  is the Bernoulli test parameter, i.e., the probability of the inliers in  $S$ . Therefore, the probability that all points in  $S$  are inliers can be expressed as follows:

$$p(I_S = m) = \prod_{x_i \in S} p(v_i = 1) \quad (12)$$

where  $v_i$  is the marker variable of whether the data point  $x_i$  is an inlier. In the absence of prior information, the probability is  $\varepsilon^m$ . Although many works (Chum and Matas, 2005; Tordoff and Murray, 2002) have realized that the probability of inliers is not necessarily the same for different data points, they still regard the probability of inliers as independent.

GroupSAC uses image segmentation or optical flow clustering

method to divide data points into groups  $\{G_i\}, i = 1 \dots K$ . The algorithm assumes that the larger the number of data points in the grouping, the higher the proportion of the inliers in the grouping. Therefore, the grouping is arranged in descending order according to the number of data points, and the grouping with more data points is sampled first. Similar to PROSAC, the sampling begins with a subset of data points with high quality, and gradually extends to all data points. In this way, the optimal model solution can be obtained more quickly, and the speed and robustness of the algorithm can be improved.

Although GroupSAC has been shown to improve sampling efficiency, its applicability depends on finding meaningful groupings in all data points. In addition, since the grouping stage is part of the robust estimation module, the specific grouping strategy (e.g., image segmentation, optical flow, etc.) should be very effective.

**HYPOTHESES VERIFICATION**

The efficiency of RANSAC algorithm largely depends on the number of outliers and the total number of data points. The larger the proportion of outliers is, the more contaminated samples are obtained. The models generated from these contaminated samples have arbitrary model parameters, which do not contain any useful information. The standard RANSAC algorithm needs to test these wrong models for all data points. Therefore, a lot of computing time is spent on this meaningless test. Some methods attempt to optimize the verification stage by reducing the time spent on evaluating error models.

Although the number of samples required by RANSAC depends on the proportion of inliers and the confidence level, the number of all data points  $N$  affects the calculation time, because the hypothesis model needs to be tested at each data point. In particular, the full run time of RANSAC can be expressed as follows:

$$t = k(t_G + Nt_v) \tag{13}$$

where  $k$  is the number of samples,  $t_G$  represents the time required to calculate model parameters, i.e., the time required to generate hypotheses, and  $t_v$  represents the time required to verify the model at data point. It can be seen from the formula that the value of  $N$  directly affects the time required for the hypotheses stage. The contaminated-free samples generated by the RANSAC algorithm are usually very few. If we assume that the consistent sets of these contaminated models are very small, we can abandon these bad models early in the verification stage. In fact, this assumption is valid in cases other than degraded data.

In recent years, many works have focused on making the verification stage of RANSAC more effective. The proposed method has the same basic idea: statistical testing for a small number of data points, and discarding or accepting models based on test results. In this case, Formula (13) can be rewritten as follows:

$$t = k(t_G + \bar{t}) \tag{14}$$

The time required to test a data point is set to unit time, and  $\bar{t}$  represents the average number of data points that a model needs to validate. The purpose of statistical testing is to reduce the number of data points that need to be tested, thus reducing the time complexity of the test stage. However, the test may mistakenly reject the "good" model, it is necessary to increase the number of sampling to maintain a certain level of confidence in the solution. Generally speaking, the more data tested, the less likely it is to reject a "good" model. So we need to find a balance between the number of test points and the number of sampling.

**R-RANSAC Based on  $T_{d,d}$  Test**

R-RANSAC (Matas and Chum, 2004) first performs a pre-test on a small subset of data points, assuming that the number of data points in this subset is  $d$ , and  $d \ll N$ . Only when all the  $d$  data points pass the test, the algorithm continues to test

the remaining  $N - d$  points, otherwise the test will be completed.

Two situations need to be considered: first, a "good" sample is extracted by probability  $P_l$ . If this sample passes the pre-test with probability of  $\alpha$ , then all  $N$  data points will be tested. Otherwise, if the sample fails to pass the pre-test with the

probability of  $1 - \alpha$ , then only  $\bar{t}_\alpha$  samples need to be tested on average; Secondly, a "bad" sample is extracted with

probability  $1 - P_l$ . If this sample passes the pre-test with probability of  $\beta$  ( $\beta \ll \alpha$ ) then all  $N$  data points will be tested. Otherwise, if the sample does not pass the pre-test with

the probability of  $1 - \beta$ , then only  $\bar{t}_\beta$  samples need to be tested on average. The average number of data points to be tested for each sample can be expressed as a function of  $d$ :

$$\bar{t}(d) = P_l(\alpha N + (1 - \alpha)\bar{t}_\alpha) + (1 - P_l)(\beta N + (1 - \beta)\bar{t}_\beta) \tag{15}$$

$\alpha$  and  $\beta$  can be calculated by the following formula:

$$\alpha = \gamma^d, \beta = \delta^d \tag{16}$$

where  $\gamma$  denotes the proportion of inliers and  $\delta$  denotes the probability that a data point is consistent with a random model. The average time of pre-test can be expressed as:

$$\bar{t}_\alpha = \sum_{i=1}^d i(1 - \gamma)\gamma^{i-1}, \bar{t}_\beta = \sum_{i=1}^d i(1 - \delta)\delta^{i-1} \tag{17}$$

The R-RANSAC algorithm derives the optimal setting of

$d = 1$  by minimizing the function  $t = k(t_G + \bar{t})$ . The  $T_{1,1}$  test first randomly extracts a data point to test the model, if the model is consistent with the data point, then continues to test; otherwise, it abandons the model and generates new samples.

As mentioned above, R-RANSAC may mistakenly reject valid hypotheses and increase the number of samples compared with standard RANSAC. However, hypotheses generation takes less time, the pre-test method can generally reduce the running time of the algorithm.

**R-RANSAC Based on SPRT Test**

In 2005, Matas and Hum proposed R-RANSAC based on Sequential Probability Ratio Test (SPRT) (Matas and Chum, 2005). This method uses likelihood ratio to determine whether

the estimated model is "good" model  $H_g$  or "bad" model  $H_b$ . The calculation formulas are as follows:

$$\lambda_j = \prod_{r=1}^j \frac{p(x_r | H_b)}{p(x_r | H_g)} = \lambda_{j-1} \cdot \frac{p(x_j | H_b)}{p(x_j | H_g)} \tag{18}$$

$p(x_r | H_b)$  and  $p(x_r | H_g)$  are conditional probabilities of observed values  $x_r$  under the assumptions of  $H_b$  and  $H_g$ ,

respectively. If the  $r$ -th data point is consistent with the model parameters, then  $x_r = 1$ , otherwise  $x_r = 0$ .  $p(1 | H_g)$  denotes the probability of randomly extracting a data point that is consistent with a "good" model. It can be approximated by the ratio of the inliers in the data points. Similarly,  $p(1 | H_b)$  denotes the probability of randomly extracting a data point consistent with a "bad" model, which can be modeled by Bernoulli distribution with parameter  $\delta$ . SPRT test is to add data points continuously and calculate the likelihood ratio  $\lambda_j$  shown in formula (18). If the ratio is greater than a threshold  $A$ , the model is considered to be a "bad" model; if the ratio does not exceed the threshold  $A$  after all data points are tested, the model is considered to be a "good" model.

Threshold  $A$  is the main parameter to be determined in SPRT test. Assuming that the parameters  $\gamma$  and  $\delta$  are known, the optimal running time can be achieved by setting  $A$ . However, these parameters are usually unknown and need to be estimated during the testing process, and the threshold  $A$  should be adjusted according to the current estimates. For example, in multi-view geometry, initial estimates of  $\delta$  can be obtained by geometric constraints; initial values of  $\gamma$  can be estimated by the maximum number of RANSAC iterations that users are willing to perform, and updated at any time using the current maximum consistent set. In the multi-view geometry experiments (Chum and Matas, 2008), R-RANSAC based on SPRT test is 2.8 to 10.9 times faster than standard RANSAC, and its efficiency in all experiments is higher than R-RANSAC based on  $T_{d,d}$  test.

## DISCUSSION AND CONCLUSIONS

As a basic step of image matching technology, mismatching point detection can not only detect mismatching points, but also make up for the shortcomings of image matching algorithm and further improve the accuracy of image matching. RANSAC method are commonly used to detect mismatched points. A number of recent research efforts have attempted to improve the performance of the standard RANSAC algorithm. In this paper, we presented a discussion and comparative analysis of a number of important RANSAC techniques.

Although there are many methods to detect mismatched points and good application results have been achieved, there are still some problems to be further studied. For example, how to find a balance between accuracy and efficiency.

## Acknowledgment

This research was supported by Taizhou Science and Technology Support (Social Development) Program (Project No. TS201701) and Taizhou Science and Technology Support (Social Development) Program (Project No. SSF20180055).

## References

- Fischler, M. A. and Bolles, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. San Francisco (CA): Morgan Kaufmann, 1987.
- Maimone, M., Cheng, Y., and Matthies, L. 2007. Two years of visual odometry on the mars exploration rovers, *Journal of Field Robotics*, 24, 169-186.
- Deigmoeller, J. and Eggert, J. 2016. Stereo visual odometry without temporal filtering, in 38th German Conference on Pattern Recognition (GCPR 2016), Hannover, Germany, 166-175.
- Torr, P. H. S. and Zisserman, A. 2000. MLESAC: A New Robust Estimator with Application to Estimating Image Geometry, *Computer Vision and Image Understanding*, 78, 138-156.
- Tordoff, B. and Murray, D. W. 2002. Guided Sampling and Consensus for Motion Estimation, in European Conference on Computer Vision, 82-98.
- Chum, O. and Matas, J. 2005. Matching with PROSAC-progressive sample consensus, in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 220-226.
- Kai, N., Hailin, J., and Dellaert, F. 2009. GroupSAC: Efficient consensus in the presence of groupings, in 2009 IEEE 12th International Conference on Computer Vision, 2193-2200.
- Matas, J. and Chum, O. 2004. Randomized RANSAC with  $T_{d,d}$  test, *Image and Vision Computing*, 22, 837-842.
- Matas, J. and Chum, O. 2005. Randomized RANSAC with sequential probability ratio test, in Tenth IEEE International Conference on Computer Vision, 1727-1732.
- Chum and Matas. 2008. Optimal randomized RANSAC, *IEEE Trans Pattern Anal Mach Intell*, 30, 1472-1482.

### How to cite this article:

Wenyan Ci *et al.* 2018, Extensions To The Standard Ransac Algorithm For Efficiency And Robustness. *Int J Recent Sci Res.* 9(12), pp. 29842-29846. DOI: <http://dx.doi.org/10.24327/ijrsr.2018.0912.2942>

\*\*\*\*\*