A Survey of RANSAC enhancements for Plane Detection in 3D Point Clouds

Article *in* Menoufia Journal of Electronic Engineering Research · July 2017

DOI: 10.21608/mjeer.2017.63627

2 authors:

Ramy Ashraf Zeineldin
Karlsruhe Institute of Technology

Nawal El-Fishawy
Faculty of Electronic Eng., Menoufia University, Menouf, Egypt

A Survey of RANSAC enhancements for Plane Detection in 3D Point Clouds

Ramy Ashraf Zeineldin and Nawal Ahmed El-Fishawy

* Dept. of Computer Science and Eng., Faculty of Elect., Eng., Menoufia University, Menouf, Egypt.

(Received: 09-March-2017 – Accepted: 29-April-2017)

Abstract

Planar surfaces are distinguished features of man-made environment, which are used in many computer vision applications such as object detection, motion segmentation, 3D scene reconstruction, and 3D mapping. One of the most used technique for robust plane detection is the RANdom SAmple Consensus (RANSAC), which is a global iterative method for estimating the parameters of a certain model from input data points contaminated by a set of outliers (noisy data). Unfortunately, the standard RANSAC suffers from some problems regarding the processing time, accuracy of fitting data, and finding an optimal solution. This paper gives a review study of the most recent RANSAC enhancements techniques. In addition, it covers the solving techniques for the speed, accuracy and optimality problems.

1. Introduction

Man-made environment consists of many parallel lines, orthogonal corners, and regular shapes such as rectangles. Therefore, planes are commonly employed, as primitive components, in various robotics and computer vision tasks. One popular application is 3D reconstruction and scene analysis [1-6], in addition, objects can be recognized accurately using 3D point cloud in [7-9], another application is motion segmentation from RGB-D videos which are represented by Bertholet et al. in [10]. Additionally, steerable displays and HoloDesk [11, 12] which allows the user to be immersed in a virtual 3D graphics environment, and finally 3D mapping from Surveillance videos are presented by Ruofei et al. [13].

The RANSAC, the most common algorithms for planar surface detection proposed by Fischler and Bolles [14], is a global parameter estimation approach designed to fit the data points contaminated by a large set of outliers, i.e. noisy data, to a predefined model.

Several methods have been proposed for enhancing the RANSAC. This study provides a survey of the most common RANSAC enhancements over the past few years [15-28]. They can be categorized according to the problems they seek to solve. It can be seen that the most important problems concerned with RANSAC enhancement are speed, accuracy and optimality.

The rest of this paper is organized sequentially as follow: The standard RANSAC approach is illustrated in Section 2. Speed, accuracy, and optimality enhancement of the standard RANSAC are discussed in Section 3. The conclusion of the different enhanced RANSAC methods is clarified in Section 4.

2. Background

2.1 Standard RANSAC

In computer vision, the RANdom SAmple Consensus (RANSAC) [14-29] is considered one of the most commonly used algorithms for plane detection. The RANSAC is firstly introduced by Fischler and Bolles [14] for 2D detection, then it has been proven by Schnabel et al. [30] to detect basic shapes, for example cones, spheres, cylinders, planes from 3D point clouds as well. The RANSAC is a global iterative method that robustly finds model parameters from a set of data points.

Figure 1 shows an example of applying RANSAC for 2D line fitting problem. By assuming data as a collection of inliers and outliers, the RANSAC can robustly estimate the parameters of planes with high degree of accuracy even number of outliers exceed 50% of the sample points. Unlike other statistical sampling techniques such as M-estimators and least-median squares [31, 32] that use as much as possible of the data, the RANSAC uses the smallest data set (starting from three points) and proceeds to enlarge this set with consistent inliers.

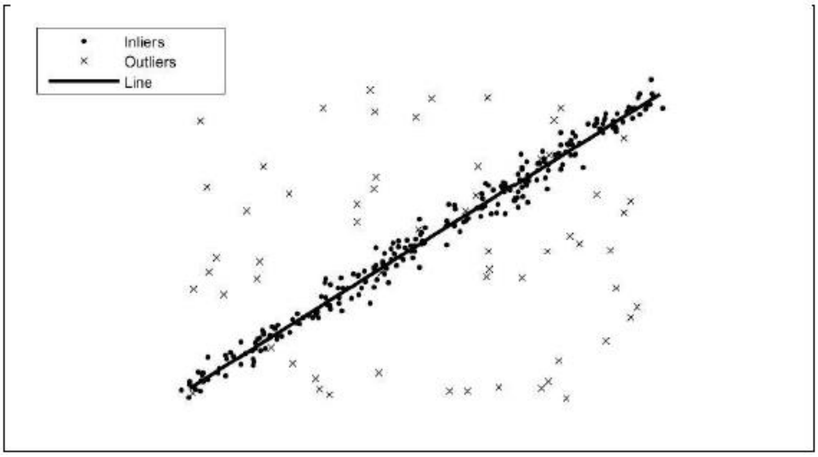

Figure 1: Estimating the line parameters using the standard RANSAC

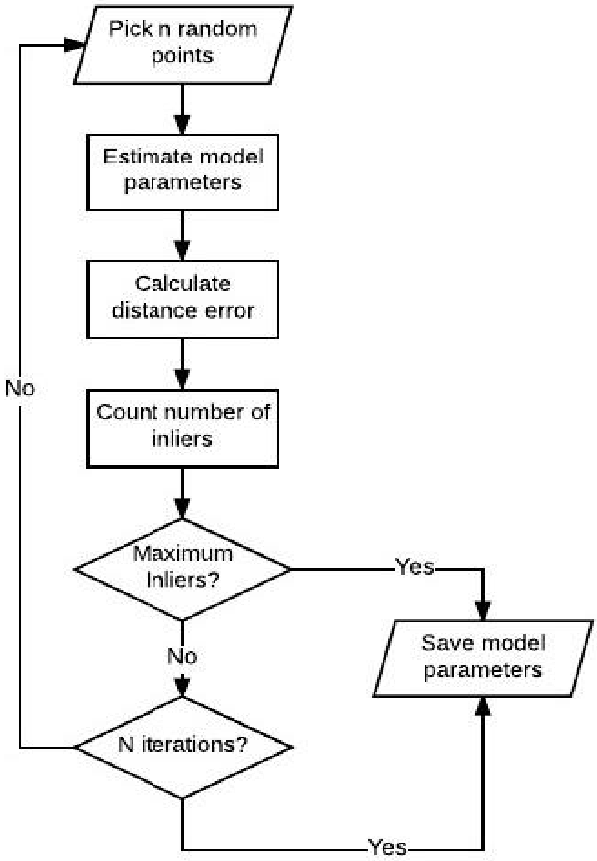

Figure 2: The RANSAC flowchart

The RANSAC algorithm consists of two simple steps that are iteratively repeated as shown in Fig. 2. Firstly, a hypothesis stage where a small collection of inliers (n), i.e. data that fits the model, is randomly selected before a fitting model and the corresponding model parameters are computed. Secondly, an evaluation stage where the RANSAC checks for data points in the entire dataset that fits the estimated model parameters obtained from the first step. If any of the input points does not fit the model with error greater than distance threshold (d), it is considered as an outlier.

2.1.1 Hypothesis stage

In the hypothesis step, the RANSAC randomly selects a subset of data points, before the parameters of the model is estimated from the input points. If the given model is plane, $Ax + By + Cz + D = 0$, and $M = [A, B, C, D]^T$ is the parameters to be estimated. Unlike common regression techniques such as least square method, the RANSAC is a resampling technique that generate candidate solutions using the smallest number of points. In other words, the RANSAC converts the estimation problem from the continuous domain to the discrete domain.

In order to obtain a good plane, the RANSAC loops for number of iterations N_{it}, which can be obtained from the following equation:

$$N_{\mathrm{it}} = \frac{\log(1-p)}{\log(1-q^S)} \tag{1}$$

Where p is the probability of finding a good plane from the input points, q is the probability that a point is an inlier, and S is the number of points in the sample.

2.1.2 Evaluation stage

The RANSAC operates in a hypothesize-and-verify phases. After the hypothesis stage, the RANSAC evaluates the candidate hypotheses to find the most suitable one, which is supported by the largest number of inlier candidates. Input data is considered inliers if only they fall below a predefined distance threshold (t), given as:

$$t = \frac{|Ax_1 + By_1 + Cz_1 + D|}{\sqrt{A^2 + B^2 + C^2}} \qquad (2)$$

Fortunately, the RANSAC does not have to extensively evaluate all the input data points, since two termination criteria can be used before that. On the one hand, the evaluation process may finish if the probability of finding a better model than the current best candidate falls below a predefined threshold. On the other hand, the termination could be achieved if the number of evaluated samples exceeds the number expected to select an uncontaminated sample.

2.1.3 RANSAC problems

The RANSAC is widely applied for estimation of a model due to its simple implementation and robustness. However, The RANSAC suffers from some common problems that are:

1. The RANSAC is a heavy computation algorithm and consumes large processing time.
2. The RANSAC fails to produce reliable results in situations with two nearby crossing planes, such as steps, stairs, curbs, or ramps, where the detected planes may contain more inliers than the real models.
3. The standard RANSAC does not find the same model in each iteration if it is applied to the same experiment. This is very important in cases where a measurement of other experiment related parameters is required.

3. The RANSAC enhancements

The RANSAC enhancement methods can be summarized as shown in Fig. 3. In the following subsections, some RANSAC's descendants that treat such problems are illustrated.

3.1 Speed Enhancement

The RANSAC's processing time can be estimated from the following formula:

$$T = S(T_H + T_E) \quad (3)$$

Where, S is number of sampled data points, T_H is time for the hypothesis generation, and T_E is time for hypothesis evaluation for each input data point.

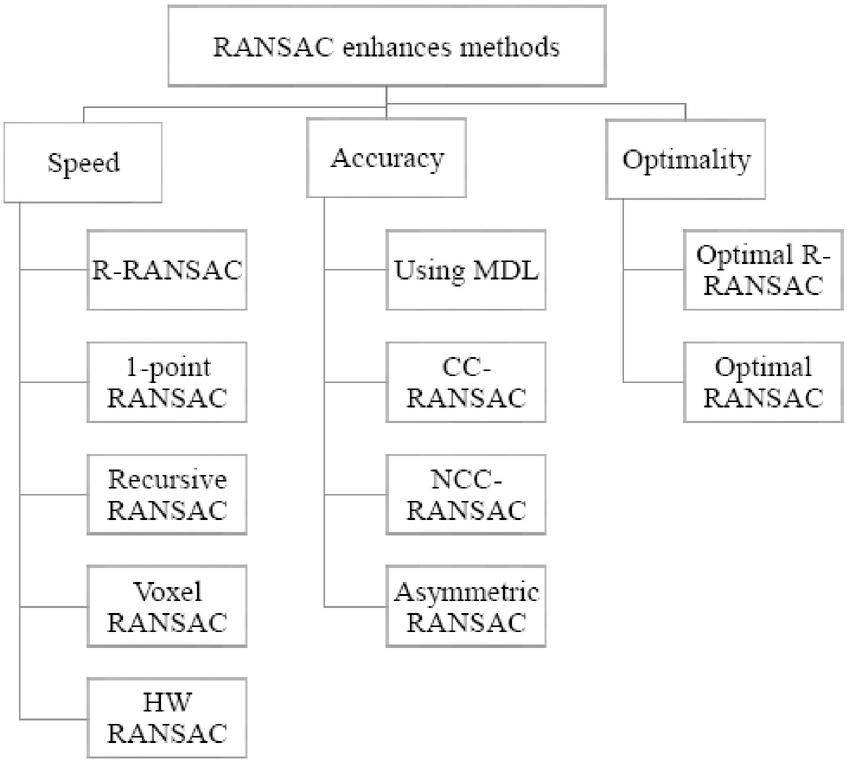

Figure 3: Classification of enhanced RANSAC methods

3.1.1 The Randomized RANSAC (R- RANSAC)

Principles- It is possible to quit the evaluation stage if the hypothesis is far from the candidate plane, which leads to reducing the evaluation time (T_E). In [15], Matas and Chum introduced a new randomized version of the RANSAC that uses this theory. The R-RANSAC (Randomized RANSAC) reduces the computing time of the evaluation step (T_E) by adding a preliminary test ($T_{d,\ d}$ Test) before the hypothesis evaluation stage.

Hypotheses is partially evaluated using a subset of the input data points d from total N points (where d<<N). This test is passed if all randomly selected d points are consistent with the candidate model. If the value of d is 0, this means standard RANSAC. However, setting d to 1 is recommended as the optimal value.

Discussion- Due to the fact that the most evaluated hypotheses are contaminated by outliers, a statistical test is performed on only a small amount of data to reject such models. Consequently, the number of verified data points is reduced (and thus the time according to equation (3)). The most important disadvantage of this method is the possibility of neglecting (bypassing) a good sample.

3.1.2 The 1-Point RANSAC

Principles- The standard RANSAC checks the consistency of data points against the global hypothesized model. The search for correspondences, or neighbor points, firstly starts with comparing local surface features to all input data points. However, applying these searching methods is very excessive and requires a lot of processing time. As a result, Civera et al. [16] introduced the 1-Point RANSAC to solve this problem by incorporating a priori probabilistic information into the hypothesis generation stage. The 1-point RANSAC combined the standard RANSAC and Extended Kalman Filter (EKF) that uses the available probabilistic knowledge before the hypothesis stage.

Discussion- Differently from the standard RANSAC, the starting point is not only input data point, but also a probability distribution over the model parameters is required. By using this prior probability information, the sample size for the hypothesis generation can be reduced to a minimum of only 1 data point. Consequently, computational savings can be achieved due to the minimization of input point and hence a smaller number of iterations according to equation (1), without the loss of discriminative power and accuracy of the RANSAC.

3.1.3 The Voxel RANSAC

Principles- In [17, 18], A preprocessing stage is proposed in order to reduce the amount of data points as much as possible while maintaining the main features. The input cloud is firstly down sampled before the hypothesis generation step of the RANSAC. This process enhances the speed of the RANSAC and avoids problems regarding this massive number of points. A voxel grid filter is applied in this filtering process.

A voxel grid is a group of small 3D cubes, or boxes, that have identical sizes. Voxel grid filter is first applied to the input data points. Then, all points in the same voxel, 3D box, are replaced with their centroid. The output cloud keeps the original geometrical information but with smaller number of points.

Discussion- The size of each cube affects the minimization of data points, in other words when the box is larger, the output cloud is smaller and vice versa. However, finding an appropriate voxel grid size that makes balance between speed and filter quality is very challenging.

3.1.4 Hardware accelerated RANSAC

Principles- Many hardware accelerating approaches are proposed to enhance the computing speed of the RANSAC algorithm. By using parallel programing based on different techniques (OpenMP, POSIX Threads, and CUDA) in [19], execution speedup is achieved. In situations where no GPUs are available, using POSIX threads is a better option than using OpenMP. Although OpenMP is easier to program, POSIX threads give the programmer more direct control to the thread primitives.

The use of Field Programmable Gate Array (FPGA) for RANSAC speed enhancement is introduced in [20-22]. Sharing the coefficient matrix in [20] not only reduces the hardware cost but also minimizes the processing time complexity. In [21], Tang et al. use double buffering mechanism for process pipelining, which implements two memory buffers for real-time operations. However, the highly parallel FPGA system in [22] excels in performance with a frame rate of 43 frame per second (fps).

Discussion- Using special hardware enhances the speed of the RANSAC to 43 fps, but this requires some additional costs and development efforts.

3.2 Accuracy Enhancement

The standard RANSAC classifies points symmetrically, i.e. a point is whether consider an inlier or outlier. If the points fulfill the plane's equation with a distance error less than threshold d, then it is considered an inlier, else an outlier. This process is iterated until the best candidate plane is found, which contains the maximum number of inliers.

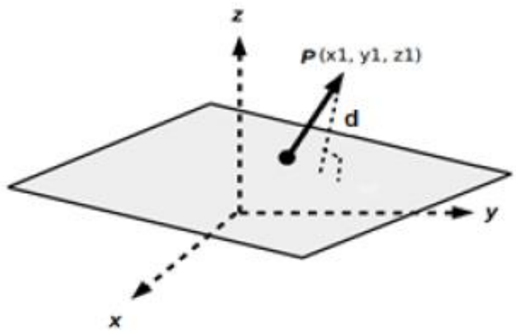

Figure 4: The perpendicular distance (d) between a point and a plane

Figure 4 illustrates the signed perpendicular distance donated by (d) from a plane Ax + By + Cz + D = 0 and a point P (x_1, y_1, z_1), can be calculated as following:

$$d = \frac{Ax_1 + By_1 + Cz_1 + D}{\sqrt{A^2 + B^2 + C^2}} \quad (4)$$

Although the RANSAC is one of the most robust algorithm for regression problem and extracting individual planar parameters, its performance is unreliable in situations when a single plane crossing two nearby patches such as steps, curbs, or ramps. In this case, the RANSAC may segment a plane intersecting with multiple planes because of seeking the largest number of inliers.

3.2.1 Integrating RANSAC and MDL

Principles- One way of eliminating such problems is by integrating it with Minimum Description Length (MDL) [23]. Firstly, the point cloud is divided into small Region Of Interests (ROIs) so that each block may include a maximum number of three planes. Then, the RANSAC is applied in each ROI to detect planes. Finally, the MDL principle is used to determine the number of planes in each ROI, which is a number from zero to three.

The principle of the MDL encoding technique can be used to present interpreting points in 3D space. For a given set of points, several hypotheses are assumed, such as outliers (O), 1 plane and outliers (1P+O), and 3 planes and outliers (3P+O).

Let n_0 points (x_i, y_i, z_i) in a 3D point cloud, the coordinates given with a resolution of ϵ and within range R. The description length for the n_0 points can be calculated given by Eq. (5), [23]:

$$\phi_0 = \#bits(po\text{int}\, s \mid O) = n_0.(3lb(R/\varepsilon)) \qquad (5)$$

Where lb(R/ϵ) are the number of bits required to describe one coordinate. If we now assume n points to sit on a plane and the other $\bar{n} = n_0 - n$ points to be outliers, we need

$$\phi 1 = \#bits(po\text{int}\, s \mid 1P + O) = n_0 + \bar{n}.3lb(R/\varepsilon) + 3lb(R/\varepsilon) + n.2lb(R/\varepsilon)$$
$$\left[\sum_{i=1}^{n}\left(\frac{1}{2\ln 2}.(x-\mu)^T \Sigma^{-1}(x-\mu) + \frac{1}{2}lb(|\Sigma|/\varepsilon^6) + \frac{k}{2}lb2\pi\right)\right] \qquad (6)$$

Where the first term is the MDL for the n_0 bits, the second term is the MDL for outliers, points not a part of the plane, and the third term represents the number of bits to describe the complexity of the plane. By assuming that n_1, n_2, n_3 inliers to randomly lie on respective planes, fourth terms are presented by applying Gaussian distribution $x \sim N(\mu, \Sigma)$.

Discussion- Integrating MDL with the RANSAC can avoid detect incorrect planes due to the complexity of 3D point clouds. The 3D data into small blocks, then a number of planes interpret each other and thus MDL is essential for deciding how many planes exists in each ROI. This approach enhances the RANSAC by not detecting the wrong plane in 3D complex scenes.

3.2.2 The Connected Component RANSAC

Principles- Figure 5 (b) shows the result of applying the standard RANSAC to scene containing a curb. Rather than detecting one plane of the three possible surfaces, the RANSAC selected a planar surface crossing all of them. This problem arises from the fact that the RANSAC chooses the plane with maximum fitness (o), i.e., the plane with the highest number of supporting data points (inliers). In order to improve the quality of measurements in such cases, Gallo et al. [24] presented dubbed Connected Component RANSAC (CC-RANSAC).

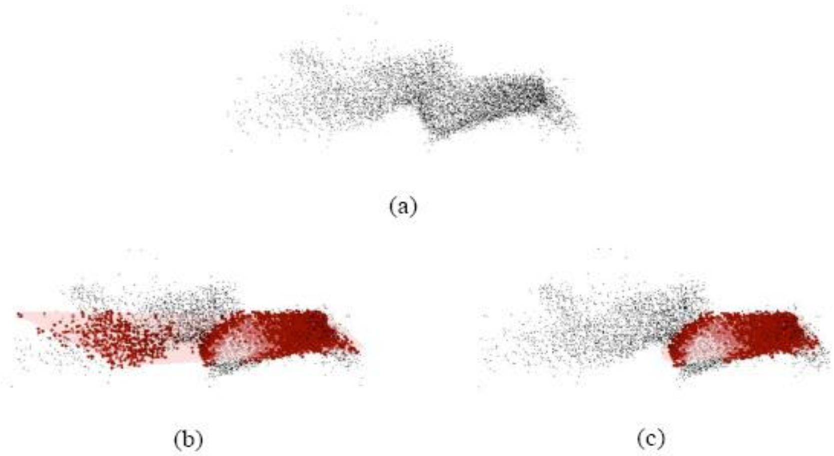

Figure 5: (a): 3D point cloud representing a curb. (b) Incorrect planar fitting using the standard RANSAC. (c) Same detected plane but using the CC-RANSAC [24].

The RANSAC makes an evaluation to the whole collection of inliers. However, the CC-RANSAC only evaluates the largest connected components of inliers. The CC-RANSAC argues that inlier points are spatially cohered and it introduces a new fitness measurement (o) as following:

$$o = |I_C(P)| \qquad (7)$$

Where, I_C is the largest surface with 8-connected component neighbors of data points. As a result of using I_C for evaluating the fitness of the candidate planes, only inliers from the lower patch is selected. The measurement acquired by a 3D camera in front of a curb are shown in a Fig. 5 (a). Standard RANSAC Incorrectly segment red points in Fig. 5 (b) because of searching for largest number of inliers.

Although there are two more planar patches visible in the scene, standard RANSAC segments the one shown in red in Fig. 5 (b) because it has the largest number of inlier data points. On the other hand, CC-RANSAC detects the points in one planar patch containing the largest number of points as shown in red in Fig. 5 (c).

Discussion-The CC-RANSAC algorithm solves the crossing-plane problem in simple scenes where planar surfaces are individual such as curbs. However, it may output inaccurate results when the input point cloud contains two or more planes at short distances close together such as stairs.

3.2.3 The NCC-RANSAC

Principles- The CC-RANSAC algorithm solves the crossing-plane problem in simple scenes where planar surfaces are individual such as curbs. However, it may output inaccurate results when the input point cloud contains two or more planes at short distances close together such as stairs. The Normal-coherence CC-RANSAC (NCC-RANSAC) approach by Qian and Ye in [25] solves this problem.

The NCC-RANSAC contains two steps. Firstly, normal coherence is estimated from RANSAC algorithm’s inliers for removing small surfaces. The output from this stage is a number of discrete patches. Secondly, rather than finding the largest connected component like the CC-RANSAC, it checks every candidate patch and performs a clustering process.

Discussion- By using both distance and normal thresholds to evaluate the candidate planes, the output planes of the NCC-RANSAC method are not over-segmented and much more accurate than the CC-RANSAC and the standard RANSAC. However, applying those additional evaluation criteria requires some additional computing cycles and the overall time is very large.

3.2.4 Asymmetric kernel

Principles- Another approach is proposed by Choi et al. [26] by applying an asymmetric kernel to the RANSAC as a score function rather than using a rectangular and symmetric kernel. The geometric distance, in equation (4), is signed so that it is positive when a point is above the plane and negative when a point is below the plane. With a priori knowledge of the ground plane parameters, error between the given point and the ground truth plane is useful to identify whether this point is an inlier or outlier. If the point is so close to the ground plane, it is considered an inlier. In contrast the point is considered an outlier or from other objects if it is far from the ground plane.

Modeling the probabilistic likelihood of a point to be an inlier or outlier can be derived in equations (8), (9) [23]. Firstly, probability of error from an inlier can be modeled using the Gaussian distribution as follows:

$$p(f_i \mid l_i = 1) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{f_i^2}{2\sigma^2}\right) \qquad (8)$$

where f_i and l_i are error and label of *i-th* point, and σ is standard deviation, that is the magnitude of inlier noise. Error from outlier can be modeled as uniform distribution as follows:

$$p(f_i \mid l_i = 0) = \begin{cases} 1/h_{\max} & if\ -h_{\max} \le f_i \le 0 \\ 0 & otherwise \end{cases} \qquad (9)$$

Where, h_{max} is the maximum allowed height for object above the ground plane.

Discussion- The asymmetric kernel is a fast approximation of the probabilistic likelihood considering inliers and outliers in a 3D point cloud which enables the RANSAC to be more robust to outliers. Although the asymmetric kernel involves more computations, its computing speed is similar to the standard RANSAC because of using less number of iterations.

3.3 Towards Optimality

One of the main disadvantage with the standard RANSAC is that it is not repeatable [33] since it is based on random sampling. In addition, instead of finding the optimal set of inliers, the RANSAC terminates when it obtains a sufficient number of inliers. As a result, the RANSAC does not perform well when a large number of contaminations exist, i.e. the number of outliers is very large. Another worse problem is the termination criteria, since the RANSAC may continue testing the data points even most inliers are found according to the defined threshold.

3.3.1 The Optimal Randomized RANSAC

Principles- The R-RANSAC mentioned in section 3.1.1, in which a two-step evaluation procedure is applied. First, a statistical test is performed on d random data points (d<<N) where N is the total data points. Then, Final evaluation on all N points is only performed if the initial test is passed. There is no upper bound on time with the R-RANSAC. Consequently, Chum and Matas described an optimal randomized RANSAC [27] based on Wald's theory of sequential decision making [34]. The hypothesis evaluation stage is assumed to be an optimization problem to decide whether the candidate model is "good" (H_g) or "bad" (H_b). Then, a "good" model is fitted to all data points.

The Wald's Sequential Probability Ratio Test (SPRT) is based on the likelihood ratio [27]:

$$\lambda_j = \prod_{r=1}^{j} \frac{p(x_r \mid H_b)}{p(x_r \mid H_g)} \qquad (10)$$

Where, x_r is equal to 1 if the r_{th} data point fits the candidate model, and 0 otherwise, p(1|Hg) represents the probability of any randomly chosen data point to be consistent with the "good" model, which is approximated by the fraction of inliers ε among the data points, and p(1|Hb) is the probability of a data point to fit a "bad" model, which is modeled as Bernoulli distribution with parameter δ [34].

For the R-RANSAC test to be optimal, the knowledge of two parameters, ε and δ is required. However, theses probabilities differ according to the data points and are assumed to be priori unknown and estimated during the sampling process. In addition, the decision threshold A is the only parameter of the SPRT which can be set to achieve optimal results with minimal average runtime to calculate the probabilities ε and δ using Wald's theorems.

Discussion- An Optimal R-RANSAC is introduced by combining adapted SPRT algorithm that remove the prior knowledge requirement of number of outliers. It has been proven to have performance close to the theoretically optimal with 2 to 10 times faster than the standard RANSAC and is up to four times faster than the R-RANSAC.

3.3.2 The Optimal RANSAC (ORANSAC)

Principles- Due to the random nature of the standard RANSAC, it is difficult to obtain the same result twice in each cycle. This is important in some applications when measuring other involved parameters such as the set of inliers. For instance, it is very important that the result does not vary while running medical applications many times in the same circumstances. Therefore, Hast et al. presented the ORANSAC to solve such problems and to make the RANSAC repeatable, i.e. gives the same result each run. Additionally, the ORANSAC provides a stopping criterion that prevents performing more data testing if the best model, containing the most inliers, is already found.

The ORANSAC resamples the input data points to a number of sets then iteratively estimate the model and the score, number of inliers, is stored in each cycle. If a large number of inliers is found in one set out from the resampling process, this is considered as a seed that begins growing by re-estimation and scoring steps. This loop will not stop until no further changes arises in this large set, which means a probable optimal solution is found. Finally, pruning stage with lower tolerance is done to keep only the most accurate inliers.

Discussion- Unfortunately, the ORANSAC might not return the same consensus twice if there is no optimal solution. This is the case when the set of inliers is very small where the probability of finding an optimal model is very rare. Another possible cause is that two discrete optimal solutions exits where S_a has one inlier more than S_b where the optimal result may fluctuate between these two solutions.

4. COMPARATIVE STUDY

Table 1 illustrates a brief comparison between the enhancements of RANSAC with respect to speed, accuracy and optimality. When the speed is very fast, it is in the range 30 fps or more, fast means 2 or more times faster than the standard RANSAC, slow indicates that the speed is similar to the standard RANSAC, whereas very slow means larger than or equal to 2 times slower than the standard RANSAC.

Regarding accuracy, it is considered low when the results is worse than the standard RANSAC, medium means a little enhancement of accuracy, additionally high is given to methods where a significant enhancement to accuracy is achieved. Finally, optimality enhances is indicated by using yes or no.

Table 1. Comparison of RANSAC enhancements [15]–[28]

N.	Method	Speed	Accuracy	Optimal?	Evaluation
1	R- RANSAC [15]	F	L	No	May reject a good model
2	1-point RANSAC [16]	VF	M	No	A prior probability knowledge of the model is required
3	Voxel RANSAC [17], [18]	VF	L	No	Determining the proper voxel grid size is very challenging
4	HW RANSAC [19]–[22]	F	H	No	Special hardware is required
5	Using MDL [23]	S	H	No	Heavy computations and complexity is very high
6	CC-RANSAC [24]	VS	M	No	Fail to produce accurate results when two patches are close together
7	NCC-RANSAC [25]	VS	H	No	The slowest
8	Asymmetric RANSAC [26]	S	M	No	Slow like RANSAC
9	Optimal R-RANSAC [27]	F	M	Yes	Fast and optimal
10	Optimal RANSAC [28]	VS	L	Yes	May not return the optimal solution if two similar solutions exist

(VF: Very Fast, F: Fast, S: Slow, VS: Very Slow)

(H: High, M: Medium, and L: Low)

5. Conclusion

This survey study covers the most vital problems of using the standard RANSAC and their presented solutions over the past years. One major problem of using the standard RANSAC is the processing speed since it is a heavy computation algorithm. Researchers proposed modifications to the standard RANSAC to minimize its intensive computations. Another crucial problem has been solved, is that despite being one of the most robust algorithms for detecting separate planar surfaces, it returns improper results when two or more crossing patches are found in the scene. Finally, the standard RANSAC quits evaluation stage when a sufficient number of relevant points are found even this is not the optimal solution. This makes the standard RANSAC not repeatable because of its random nature, some approaches have solved the optimality problem in certain circumstances

References

[1] Y. M. Kim, N. J. Mitra, D.-M. Yan, and L. Guibas, "Acquiring 3D indoor environments with variability and repetition," ACM Trans. Graph., vol. 31, no. 6, p. 1, 2012.

[2] C. V. Nguyen, S. Izadi, and D. Lovell, "Modeling kinect sensor noise for improved 3D reconstruction and tracking," Proc. - 2nd Jt. 3DIM/3DPVT Conf. 3D Imaging, Model. Process. Vis. Transm. 3DIMPVT 2012, pp. 524–530, 2012.

[3] M. Nie\ssner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3D Reconstruction at Scale Using Voxel Hashing," ACM Trans. Graph., vol. 32, no. 6, pp. 169:1–169:11, 2013.

[4] M. Zollh, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, A. Fitzgibbon, C. Loop, C. Theobalt, and M. Stamminger, "Real-time Non-rigid Reconstruction using an RGB-D Camera," 2013.

[5] A. Dai, M. Nießner, M. Zollhöfer, and … S. I., "BundleFusion: Real-time Globally Consistent 3D Reconstruction using On-the-fly Surface Re-integration," arXiv.org, 2016.

[6] M. Innmann, M. Zollhöfer, M. Nießner, C. Theobalt, and M. Stamminger, "VolumeDeform: Real-time Volumetric Non-rigid Reconstruction," pp. 1–17, 2016.

[7] L. Alexandre, "3D Object Recognition using Convolutional Neural Networks with Transfer Learning between Input Channels," 13th Int. Conf. Intell. Auton. Syst., 2014.

[8] S. Aigerim, A. Askhat, and A. Yedilkhan, "Recognition of 3D object using Kinect," Appl. Inf. Commun. Technol. (AICT), 2015 9th Int. Conf., pp. 341–346, 2015.

[9] G. Pang and U. Neumann, "Fast and Robust Multi-view 3D Object Recognition in Point Clouds," 3D Vis. (3DV), 2015 Int. Conf., pp. 171–179, 2015.

[10] P. Bertholet, A.-E. Ichim, and M. Zwicker, "Temporally Consistent Motion Segmentation from RGB-D Video," 2016.

[11] O. Hilliges, Kim, S. Izadi, M. Weiss, and A. Wilson, "Holodesk: Direct 3d interactions with a situated see-through display," Proc. CHI 2012, pp. 2421–2430, 2012.

[12] A. Wilson, H. Benko, S. Izadi, and O. Hilliges, "Steerable augmented reality with the beamatron," UIST '12 Proc. 25th Annu. ACM Symp. User interface Softw. Technol., pp. 413–422, 2012.

[13] R. Du, "Video Fields : Fusing Multiple Surveillance Videos into a Dynamic Virtual Environment," pp. 1–8, 2016.

[14] M. a. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," Commun. ACM, vol. 24, no. 6, pp. 381–395, 1981.

[15] J. Matas and O. Chum, “Randomized RANSAC with Td,d test,” Image Vis. Comput., vol. 22, no. 10 SPEC. ISS., pp. 837–842, 2004.
[16] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, “1-Point RANSAC for EKF Filtering. Application to Real-Time Structure from Motion and Visual Odometry,” J. F. Robot., vol. 27, no. 5, pp. 609–631, 2010.
[17] P. Gyawali and J. McGough, “Simulation of detecting and climbing a ladder for a humanoid robot,” IEEE Int. Conf. Electro Inf. Technol., 2013.
[18] J. Pardeiro, J. V. Gómez, D. Álvarez, and L. Moreno, “Learning-based floor segmentation and reconstruction,” Adv. Intell. Syst. Comput., vol. 253, pp. 307–320, 2014.
[19] A. Hidalgo-Paniagua, M. A. Vega-Rodríguez, N. Pavón, and J. Ferruz, “A Comparative Study of Parallel RANSAC Implementations in 3D Space,” Int. J. Parallel Program., vol. 43, no. 5, pp. 703–720, 2014.
[20] L. Dung, C. Huang, and Y. Wu, “Implementation of RANSAC Algorithm for Feature-Based Image Registration,” J. Comput. Commun., vol. 2013, no. November, pp. 46–50, 2013.
[21] J. W. Tang, N. Shaikh-Husin, and U. U. Sheikh, “FPGA implementation of RANSAC algorithm for real-time image geometry estimation,” Proceeding - 2013 IEEE Student Conf. Res. Dev. SCOReD 2013, no. December, pp. 290–294, 2015.
[22] J. Vourvoulakis, J. Lygouras, and J. Kalomiros, “Acceleration of RANSAC algorithm for images with affine transformation,” 2016 IEEE Int. Conf. Imaging Syst. Tech., pp. 60–65, 2016.
[23] M. Y. Yang and W. Förstner, “Plane Detection in Point Cloud Data,” Proc. 2nd Int. Conf. Mach. Control Guid. Bonn, no. 1, pp. 95–104, 2010.
[24] O. Gallo, R. Manduchi, and A. Rafii, “CC-RANSAC: Fitting planes in the presence of multiple surfaces in range data,” Pattern Recognit. Lett., vol. 32, no. 3, pp. 403–410, 2011.
[25] X. Qian and C. Ye, “NCC-RANSAC : A Fast Plane Extraction Method for Noisy Range Data,” IEEE Trans. Cybern., vol. 44, no. 12, pp. 2771–2783, 2014.
[26] S. Choi, J. Park, J. Byun, and W. Yu, “Robust Ground Plane Detection from 3D Point Clouds,” in Proceedings of the 2014 14th International Conference on Control, Automation and Systems (ICCAS 2014), 2014, pp. 1076–1081.
[27] O. Chum and J. Matas, “Optimal randomized RANSAC,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 30, no. 8, pp. 1472–1482, 2008.
[28] a Hast and J. Nysjö, “Optimal RANSAC-Towards a Repeatable Algorithm for Finding the Optimal Set,” Wscg, vol. 21, pp. 21–30, 2013.
[29] J. C. McGlone, E. M. Mikhail, J. S. Bethel, R. Mullen, and American Society for Photogrammetry and Remote Sensing., Manual of

photogrammetry. American Society for Photogrammetry and Remote Sensing, 2004.
[30] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for point-cloud shape detection," Comput. Graph. Forum, vol. 26, no. 2, pp. 214–226, 2007.
[31] C. V. Stewart, "Robust Parameter Estimation in Computer Vision," SIAM Rev., vol. 41, no. 3, pp. 513–537, 1999.
[32] P. J. Huber, Robust Statistics, vol. 82, no. 3. 1981.
[33] M. Zuliani, "RANSAC for Dummies," 2012.
[34] A. Wald, Sequential Analysis. Dover, 1947.

مسح لتحسينات توافق العينات العشوائية للكشف عن المستويات في الصور ثلاثية الأبعاد

الأسطح المستوية هي السمات المميزة للبيئة من صنع الإنسان، والتي تستخدم في العديد من تطبيقات الرؤية بالحاسب مثل تحديد الوجوه، وتجزئة الحركة، وإعادة بناء المشهد ثلاثي الأبعاد، وبناء الخرائط ثلاثية الأبعاد، وتعد هذه التقنية (RANSAC أو توافق العينات العشوائية) الأكثر استخداما للكشف عن الأسطح المستوية، وهو استخدام أسلوب التكرار العشوائي لتقدير معالم نموذج معين من نقاط البيانات المدخلة التي تحتوي مجموعة من القيم الدخيلة (البيانات الصاخبة). وللأسف، فإن توافق العينات العشوائية القياسية يعاني من بعض المشاكل، على سبيل المثال أنها مكلفة للغاية من حيث وقت المعالجة. مشكلة أخرى هي عدم حصول طريقة توافق العينات العشوائية إلى نتائج دقيقة الانتاج في الحالات التي يكون فيها وجود مستوى قاطع لمستويين أخرين. ويرجع ذلك إلى طريقة التركيب الأساسية المعتمدة على المسافة، والبيانات الناتجة تحتوي نقاط من المستويين معا بدلا من الحصول على مستوى واحد فقط. وأخيرا، فإن توافق العينات العشوائية القياسية يكون سيء للغاية عندما يكون عدد النقاط التي تنتمي للمستوي هو أقل من ٥٠٪. لذلك قام العديد من الباحثين على مدى السنوات الماضية بتشكيل الكثير من التحسينات لتوافق العينات العشوائية لحل مثل هذه المشاكل. في هذه الدراسة تم عرض معظم هذه التقنيات المحسنة، وفقا للمشكلة التي تم حلها. كما تم عمل مقارنة تهدف إلى تحديد القيم (السرعة، الدقة، الكمالية) لكل من الطرق المذكورة.