Volume 9, Number 1, 2023

Energy Efficient RANSAC Algorithm for Flat Surface Detection in Point Clouds

Anatoliy Zhuchenko, Oleksiy Kuchkin, Artem Sazonov*, Danylo Zghurskyi

National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37 Peremohy Avenue, Kyiv, 03056, Ukraine

Received: March 30, 2023. Revised: May 29, 2023. Accepted: June 08, 2023.

Abstract

Mobile robots control systems achieve greater efficiency through the use of robust environmental analysis algorithms based on data collected from optical sensors such as depth cameras, Light Detection and Ranging sensors (LIDARs). These data sources provide information about control object environment in point cloud. The work of such algorithms, as a rule, is aimed at detecting the objects of interest and searching for the specified objects, as well as relocating its own position on the scene. There are many different approaches for solving object detection problem in point clouds, but most of them require high computational resources. In this work, many variations of the random sample consensus (RANSAC) method are analyzed for objects defined by a mathematical model of an analytical form. Statistical characteristics of data analysis were used to compare the methods. The results demonstrate the most energy efficient flat surface detection method that processes 60 RGB-D camera frames per second.

Keywords: RANSAC; plane detection; point cloud; UAV.

1. Introduction

The detection problem and primary analysis of environmental objects are relevant for the mobile robots automated control systems, since the result of solving this problem is the basis for the robot localization and navigation. Such systems should be the most resource-efficient and at the same time provide effective control. The map determination surrounding robot and object analysis can be ensured by LIDAR. Optical systems which are based on the phenomena of light reflection usually demonstrate high accuracy in the representing of environment 3D map [1], [2], but have many erroneous indicators due to uneven scattering of light in the environment [3]. Therefore, for the synthesis of a high-precision control system, it is necessary to use an optimal algorithm for detecting objects from a 3D sparse map of the environment.

AI-based methods, primarily neural networks, cannot be used for this task [4], [5], as the inference time is much longer than the time it takes to get new data from the sensors, which means that it is impossible to analyze these data using the neural network in real time (more than 30 frames per second). This problem can be also solved in the other way, one of them is Random Sample Consensus (RANSAC) method. RANSAC algorithm has numerous modifications which can adapt it to the specific tasks solving as well as significantly improves its properties: algorithm complexity, speed, accuracy, robustness, etc. Thus, the aim of paper is to analyze the existing RANSAC family methods, among which: Basic RANSAC implementation (BASE); R-RANSAC with Sequential Probability Ratio Test (SPRT); R-RANSAC with Tdd test (TDD); Maximum Likelihood SAC (MLE); Progressive SAC (PRO); RANSAC with L-estimator (LSE); Genetic Algorithm SAC (GA), for plane detection and to choose the most optimal one that can work in real time.

* Corresponding author. Email address: a.sazonov@kpi.ua

This paper should be cited as: A. Zhuchenko, O. Kuchkin, A. Sazonov, D. Zghurskyi. Energy efficient RANSAC algorithm for flat surface detection in point clouds. *Energy Engineering and Control Systems*, 2023, Vol. 9, No. 1, pp. 47 – 53. https://doi.org/10.23939/jeecs2023.01.047

2. Related work

Analysis of objects on a sparse 3D map is performed using the body geometric and optical characteristics. The logical representation of the body of a certain set of points μ_i that need to be separated from the general set of points M can be presented as a set of certain three-dimensional primitives Ω, and written as a system of equations, where each equation σ_i describe a geometric primitive (line, plane, parallelepiped, sphere, etc.). Optical characteristics (color and intensity) can be written as a set of constraints Υ to the system Ω in the form of inequalities. Thus, a mathematical model for objects that need to be detected on a certain set of points is M_i given as a system of equations with constraints.

The search process by the RANSAC algorithm consists of two steps: generating a statistical hypothesis with a given confidence probability; testing the hypothesis on the required mathematical model [6]. These steps are performed iteratively until the desired result accuracy or other algorithm stop criterion is reached.

Variations of the original RANSAC method, as noted above, can improve its properties for different types of problems and mathematical models [7]. Typically, RANSAC family algorithms are grouped by application: with specific features: accurate, fast, robust; and their performance is estimated by fitting the model to data and estimating planar homography [8]. The adaptations of the random sampling method considered in section III are investigated to select the optimal detection algorithm on board the UAV, therefore the most important criteria for evaluating the method are the speed of its work and the algorithmic complexity of calculating the algorithm. The paper does not consider the methods of additional optimization of RANSAC algorithms by parallelization or vectorization of calculations, since the automatic control system also consists of other software blocks, and spends the main resources on board the mobile robot.

3. Applied methods overview

In this work, variations of RANSAC algorithms for planes detection in a sparse map of points are analyzed. The mathematical model for hypothesis testing looks like this:

$$\begin{cases} \sigma_i: A_i x_s + B_i y_s + C_i z_s + D_i = 0, \\ \mu_i: A_i x_t + B_i y_t + C_i z_t + D_t < threshold, \\ \vdots \\ P(v_i) \geq 0.95 \\ i \geq 1 \end{cases}, \tag{1}$$

where i is the detected plane number; σ_i is the plane equation; μ_i is the subset of points that define the ith plane; $P(v_i)$ is the confidence probability; $threshold$ is maximum distance from a point to a plane at which it is defined as inlier.

Stopping criterion for plane searching in point cloud:

$$(1 - \epsilon_i^3)^k < 1 - P(v_i)\,, \tag{2}$$

where k is the number of previous iterations; ϵ_i is the ratio of the number of subset points μ_i to the total number of points in cloud M.

Therefore, the basic (base) variation of random sample consensus method, considered by us, has an algorithm corresponding to the original paper [9]:

Set: $k \leftarrow 0,\ \mu_i \leftarrow \emptyset, l_i \leftarrow size(\mu_i),\ \epsilon_i \leftarrow 3/size(M)$

1 Sample set $\lambda_j, j = 1,2,3$ from M randomly;

2 Estimate model σ_i parameters (A_i, B_i, C_i, D_i) using sample data λ_j;

3 Compute inliers μ_k and count them $l_k \leftarrow size(\mu_k)$

4 Update inliers $\mu_i \leftarrow \mu_k, l_i \leftarrow l_k$ and error $\epsilon_i \leftarrow l_k/size(M)$ if $l_k > l_i$

5 Repeat 1–4 until not (2) and increase iteration variable $k \leftarrow k + 1$

Algorithm 1: The basic RANSAC algorithm for objects detection specified by model (1)

3.1. RANSAC variations based on partial evaluation

Increasing the surface detection speed of the random sampling algorithm by reducing the test sample at stage 3 (Algorithm 1) can significantly increase the efficiency of the algorithm, with a slight decrease in the accuracy of its operation. The most popular variations using partial model estimation of RANSAC algorithm are the T_{dd} and SPRT methods. The method of preliminary estimation of the model (T_{dd}) [10], uses a much smaller sample of data for testing and in the case of passing preliminary testing, the hypothesis generated in stage 2 (Algorithm 1) is tested on the full set of points. Another approach, R-RANSAC with SPRT, is based on successive evaluation of small subsamples of the point cloud, an iterative method until the probability ratio of the hypothesis is lower than a certain, given threshold [11]. The general algorithm using the property of partial estimation of the model can be presented as follows:

Set test set: $Q \leftarrow q_i \in M, size(Q) \ll size(M)$

Set: $k \leftarrow 0,\ \mu_i \leftarrow \emptyset, l_i \leftarrow size(\mu_i),\ \epsilon_i \leftarrow 3/size(M)$

1 Sample set $\lambda_j, j = 1,2,3$ from M randomly;

2 Estimate model σ_i parameters (A_i, B_i, C_i, D_i) using sample data λ_j;

3 Compute inliers μ_k from Q and count them $l_k \leftarrow size(\mu_k)$

4 Update inliers $\mu_i \leftarrow \mu_k, l_i \leftarrow l_k$ and error $\epsilon_i \leftarrow l_k/size(Q)$ if $l_k > l_i$

5 Compute inliers μ_M from M if hypothesis is accepted

6 Repeat 1–4 until not (2) and increase iteration variable $k \leftarrow k + 1$

Algorithm 2: Algorithm of RANSAC variations based on partial hypothesis evaluation

3.2. Guided sampling RANSAC methods

The RANSAC algorithm can be improved by guided sampling for hypothesis generation/testing. Such methods are aimed at increasing the convergence speed of the RANSAC algorithm. The main idea of increasing the efficiency, introducing the cost function and a priori probabilities for the global point cloud, and selecting the data to generate the hypothesis λ_j non-randomly, in step 1 (Algorithm 1). However, these modifications of the method make it slow due to the additional computational load of global search in point cloud M. The most well-known methods are based on guided data sampling: MLESAC [12] and PROSAC [13]. Both methods use a semi-random algorithm to generate a hypothesis, but the Progressive algorithm calculates a preliminary estimate of the coincidence of the planar homography and sorts points in cloud M for subsequent iterations. A guided sampling algorithm that modifies base RANSAC method:

Set: $P(M) \leftarrow \{1/size(M)\}$ – prob. of choosing i point

Set: $k \leftarrow 0,\ \mu_i \leftarrow \emptyset, l_i \leftarrow size(\mu_i),\ \epsilon_i \leftarrow 3/size(M)$

1 Sample set $\lambda_j, j = 1,2,3$ from M based on $P(M)$;

2 Estimate model σ_i parameters (A_i, B_i, C_i, D_i) using sample data λ_j;

3 Compute inliers μ_k and count them $l_k \leftarrow size(\mu_k)$

4 Update inliers $\mu_i \leftarrow \mu_k, l_i \leftarrow l_k$ and error $\epsilon_i \leftarrow l_k/size(M)$ if $l_k > l_i$

5 Update priors $P(M)$ if hypothesis is accepted

6 Repeat 1–5 until not (2) and increase iteration variable $k \leftarrow k + 1$

Algorithm 3: Sampling strategy for the sample consensus algorithm

3.3. Adaptations to improve accuracy and robustness

The accuracy of RANSAC algorithm corresponds to the ratio of the number of matches μ_k to the valid values of the model $\mathcal{M}$. The robustness of the algorithm, on the contrary, is determined by the number of incorrectly determined matches. Methods that allow increasing the accuracy and robustness of the algorithm use the general idea of refining the model at each iteration of the algorithm. The genetic algorithm (GA) and Least Squares (LSE) adaptations are most often used, which have a rather strong impact on the computational complexity, but provide high accuracy. The GA for the random sampling method has a relatively unique algorithm [14] because it manages the data set as a hypothesis-generating genome. In the case of a poorly generated hypothesis, the genome receives a penalty that reduces its chances for development. LSE adaptation performs model fitting based on mathematical optimization at each iteration of the method [15]. Algorithms can be written as follows:

Set: $k \leftarrow 0,\ \mu_i \leftarrow \emptyset, l_i \leftarrow size(\mu_i),\ \epsilon_i \leftarrow 3/size(M)$

1 Sample set $\lambda_j, j = 1..d$ from M randomly, $d \geq 3$;

2 Estimate model/models σ_i parameters (A_i, B_i, C_i, D_i) using sample data λ_j;

3 Compute inliers $\mu_{k,model}$ and count them $l_{k,model} \leftarrow size(\mu_{k,model})$

4 Choose the best model or optimize based on inliers $\mu_i \leftarrow best(\mu_{k,model}), l_i \leftarrow best(l_{k,model})$

5 Update inliers $\mu_i \leftarrow \mu_k, l_i \leftarrow l_k$ and error $\epsilon_i \leftarrow l_k/size(M)$ if $l_k > l_i$

6 Repeat 1-5 until not (2) and increase iteration variable $k \leftarrow k + 1$

Algorithm 4: A general algorithm for increasing the accuracy of the RANSAC method

4. Evaluation

Evaluation of methods for detecting planes is performed on the basis of comparison of performance characteristics of algorithms and visual estimation of planar homography of the result. For each data set, 100 algorithm tests were performed, and RMS values of the evaluation criteria (Fig. 1-2).

Table 1. Comparison of variations of the RANSAC method for plane detection by the number of iterations.

Failure rate	Method Abbr.	Data1	Data2	Data3	Data4	Data5	Data6	Data7	Data8	Data9	Data10	Mean
10%	BASE	7	7	7	7	7	7	7	7	7	7	7
	TDD	7	7	7	7	7	7	7	7	7	7	7
	SPRT	9	8	9	8	9	9	9	9	9	9	9
	GA	7	6	6	7	7	7	7	7	6	7	7
	PRO	7	7	7	7	6	7	7	7	7	7	7
	MLE	**4**	**4**	**4**	**4**	**4**	**4**	**5**	**4**	**4**	**5**	**4**
	LSE	6	6	6	6	6	6	6	6	6	6	6
5%	BASE	8	8	8	8	8	8	8	8	8	8	8
	TDD	8	8	8	7	8	8	8	8	8	8	8
	SPRT	10	11	10	11	11	10	11	12	11	11	11
	GA	8	8	8	8	7	7	7	8	7	8	8
	PRO	8	7	7	7	8	8	7	8	8	7	8
	MLE	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**6**	**5**	**5**	**5**
	LSE	7	7	7	7	7	7	7	7	7	7	7

Table 2. Comparison of variations of the RANSAC method for plane detection by search time [ms].

Failure rate	Method Abbr.	Data1	Data2	Data3	Data4	Data5	Data6	Data7	Data8	Data9	Data10	Mean
10%	BASE	53	49	46	48	45	45	46	49	47	48	48
	TDD	19	18	19	18	18	19	19	19	18	19	19
	SPRT	**15**	**15**	**17**	**16**	**16**	**17**	**16**	**18**	**16**	**16**	**16**
	GA	328	297	289	294	288	285	286	294	286	291	294
	PRO	118	101	106	98	95	97	99	99	106	100	102
	MLE	146	137	127	127	129	125	134	119	126	137	131
	LSE	67	75	76	76	76	78	74	75	73	73	74
5%	BASE	62	68	61	69	64	60	62	64	57	55	62
	TDD	22	23	22	22	23	22	23	24	23	22	23
	SPRT	**18**	**20**	**18**	**19**	**18**	**18**	**19**	**20**	**19**	**18**	**19**
	GA	359	390	365	419	382	354	348	363	330	330	364
	PRO	132	135	125	140	137	126	123	127	118	114	127
	MLE	173	194	160	197	177	173	162	184	151	157	173
	LSE	78	79	77	82	78	85	83	84	86	82	81

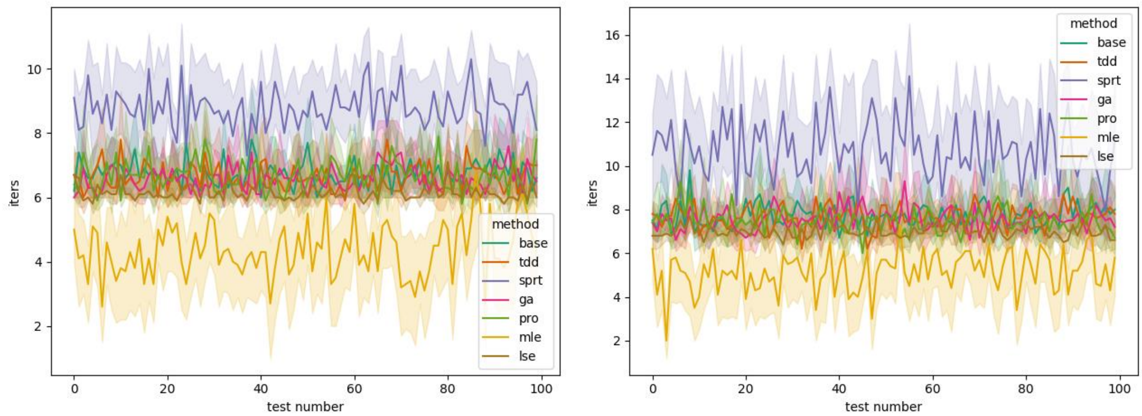
We use quantification for variations of RANSAC method, using self-assembled, Microsoft Kinetic sensor-based datasets. The datasets are presented as a reconstructed plane (1 [m^2]) with many small objects placed on top of the plane. Thus, the total complexity of datasets is 200–300 thousand points that including one plane. The confidence probabilities of the search algorithms are chosen to be 90% and 95%. Increasing confidence probabilities, for selected datasets, does not lead to increase the accuracy, but reduces the performance of the methods.

4.1. Algorithm validation

For the validation of RANSAC methods, two evaluation criteria were used: the number of iterations; plane search time. Table 1 and Table 2 demonstrate the full effectiveness of the methods according to their average values of the two criteria, respectively. All calculations are performed on Jetson NANO in Python v3.11 environment with ARM Cortex-A57 MPCore @ 1.4 GHz.

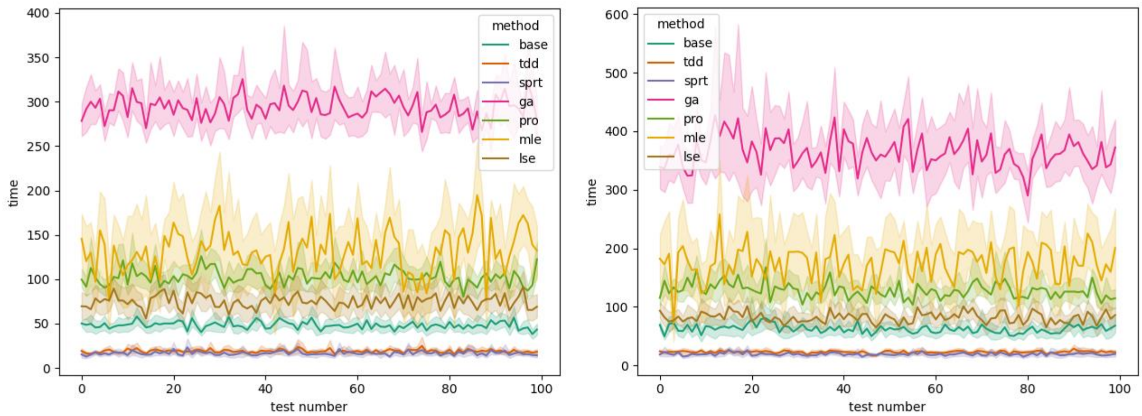
4.2. Qualitative comparison of methods

The choice of the most optimal method is based on the possibility of using it to analyze three-dimensional point clouds in real time. Thus, MLESAC (by the number of iterations, see Table 1) and R-RANSAC with SPRT (by search time, see Table 2) can be considered the best for our problem. To check the stability of the algorithms, the root mean square deviations of the two criteria for different confidence intervals are also presented (Fig.1, Fig.2).



a *b*

Fig.1. Iteration number standard deviation for 10% (*a*) and 5% (*b*) failure rate.



a *b*

Fig.2. Computation time standard deviation for 10% (*a*) and 5% (*b*) failure rate.

According to the corresponding graphs of criteria values, it can be concluded that the R-RANSAC with SPRT method has significantly smaller criteria deviations. This method shows more stable search time and number of iterations, unlike MLESAC. And, for a more detailed analysis, histograms of the number of tests in the range of criteria were constructed, demonstrating their nature of the distribution of RMS values (Fig.3).

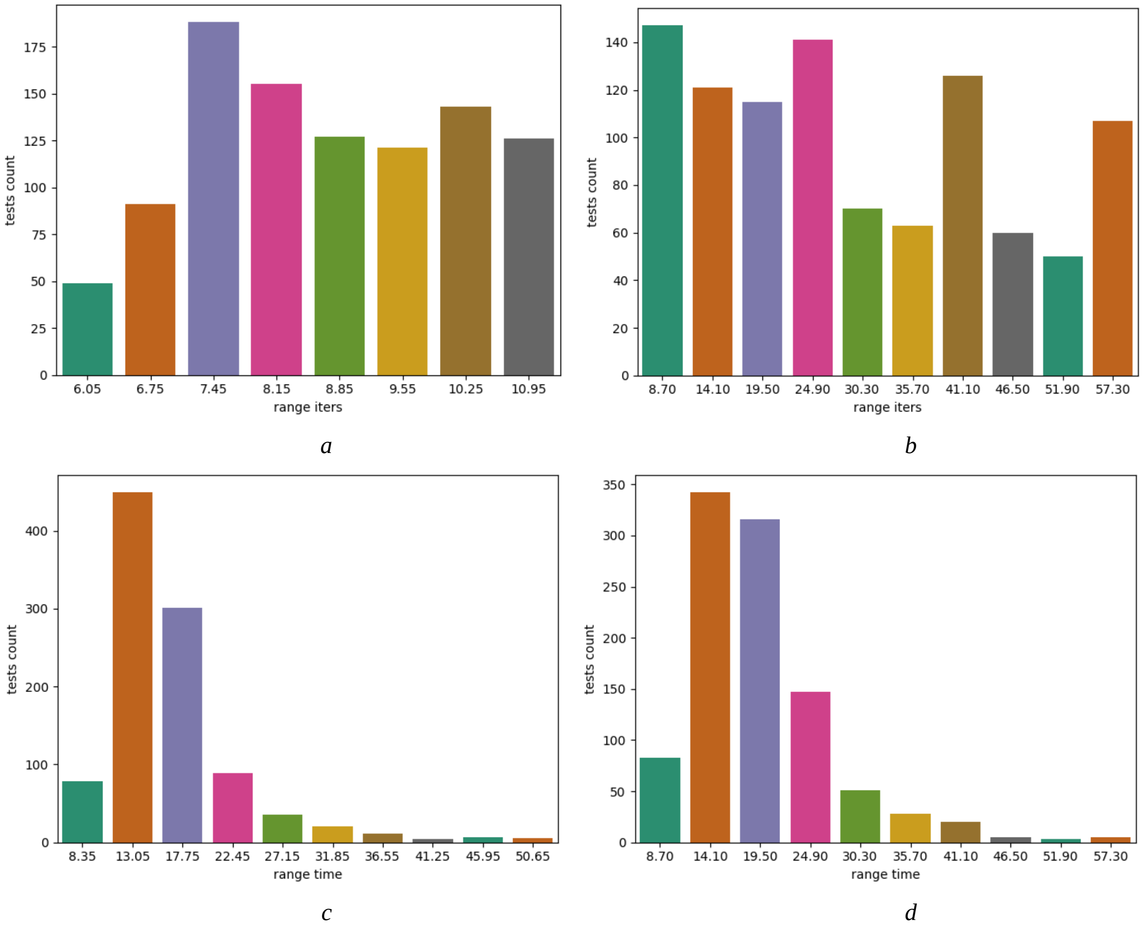


Fig.3. Standard deviation histograms of iterations (*a*, *b*) and computation time (*c*, *d*) for 10% (*a*, *c*) and 5% (*b*, *d*) failure rate based on RANSAC with SPRT for 100 tests of 10 datasets.

5. Conclusion

This paper analyzes, the methods of plane detection in point clouds using the random sample consensus algorithm. For the actual problem of autonomous navigation, modifications of the algorithm for planes detection in point cloud are considered, and the properties of the implementation and operation of each method for a deep understanding of their effectiveness are determined. Statistical characteristics for differentiated data sets are given in Table 1 and Table 2 are presented taking into account the evaluation of the planar homography of the detection result so that the deviation of the separated objects points subsets for various modifications of the method are minimal. And for the selected modifications of the algorithm, an increased number of tests were carried out, given the random nature of the method. Therefore, the obtained metrics of the methods accurately reflect their properties of working on real data.

Results show that the most efficient algorithm of the RANSAC family that can be used for real-time plane detection is a method based on partial estimation – RANSAC with a sequential probability ratio test. As can be seen from the obtained quantitative characteristics of the criteria, the method converges in 8..57 iterations for 5% failure rate, and in 6..10 iterations for 10% failure rate, respectively. The convergence time of the method has a uniquely normal distribution with average values of 15[ms] for 5% failure rate, and in 15[ms] for 10% failure rate, respectively. Thus, it can be calculated that real-time frame processing will run at 60[fps] with a drop to 30[fps] in 5% of all frames analyzed.

6. Acknowledgment

This paper has been created in the context of the MASTERLY project GA n.101091800.

References

[1] J. Ren, K. McIsaac, and R. Patel, “Modified newton’s method applied to potential field-based navigation for mobile robots,” IEEE Transactions on Robotics, vol. 22, no. 2, pp. 384–391, 2006.

[2] Aldao, E.; González-de Santos, L.M.; González-Jorge, H. LiDAR Based Detect and Avoid System for UAV Navigation in UAM Corridors. Drones 2022, 6, 185. https://doi.org/10.3390/ drones6080185.

[3] Matous Vrba, Viktor Walter and Martin Saska. On Onboard LiDAR-based Flying Object Detection. 9 Mar 2023. https://arxiv.org/pdf/2303.05404.pdf

[4] Abhijeet Shenoi, Mihir Patel, JunYoung Gwak, Patrick Goebel, Amir Sadeghian, Hamid Rezatofighi, Roberto Mart´ın-Mart´ın, Silvio Savarese. JRMOT: A Multi-Modal Real-Time 3D Multi-Object Tracker and a New Large-Scale Dataset. 22 Jul 2020. https://arxiv.org/pdf/2002.08397.pdf

[5] Jinze Liu, Minzhe Li, Jiunn-Kai Huang, Jessy W. Grizzle. Realtime Safety Control for Bipedal Robots to Avoid Multiple Obstacles via CLF-CBF Constraints. 5 Jan 2023. https://arxiv.org/pdf/2301.01906.pdf

[6] Jiaqi Yang, Zhiqiang Huang, Siwen Quan, Qian Zhang, Yanning Zhang, Senior Member, IEEE and Zhiguo Cao. On Efficient and Robust Metrics for RANSAC Hypotheses and 3D Rigid Registration.10 Nov 2020. https://arxiv.org/pdf/2011.04862.pdf

[7] Sunglok Choi1, Taemin Kim2, Wonpil Yu1. Performance Evaluation of RANSAC Family. BMVC 2009 doi:10.5244/C.23.81

[8] Bin Tan, Nan Xue, Tianfu Wu, Gui-Song Xia. NOPE-SAC: Neural One-Plane RANSAC for Sparse-View Planar 3D Reconstruction. 30 Nov 2022. https://arxiv.org/pdf/2211.16799.pdf

[9] Fisher, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography; Comm. of the ACM 24(6), 381-395, 1981.

[10] Jiri Matas and Ondrej Chum. Randomized RANSAC with Td,d test. Image and Vision Computing, 22(10):837–842, September 2004.

[11] J. Matas; O. Chum. Randomized RANSAC with sequential probability ratio test. 05 December 2005 doi: 10.1109/ICCV.2005.198

[12] Torr, P.; Zisserman, A. MLESAC: A New Robust Estimator with Application to Estimating Image Geometry. Comput. Vis. Image Underst. 2000, 78, 138–156, doi:10.1006/cviu.1999.0832.

[13] Chum, O.; Matas, J. Matching with PROSAC—Progressive sample consensus. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), San Diego, CA, USA, 20–26 June 2005; Volume 1, pp. 220–226.

[14] Ehsan Shojaedinia, Mahshid Majda, Reza Safabakhsha. Novel Adaptive Genetic Algorithm Sample Consensus. 26 Nov 2017. https://arxiv.org/pdf/1711.09398.pdf

[15] Zhang, Qingming, Buhai Shi, and Haibo Xu. 2019. "Least Squares Consensus for Matching Local Features" *Information* 10, no. 9: 275. https://doi.org/10.3390/info10090275

Енергоефективний RANSAC алгоритм для детектування площин в хмарі точок

Анатолій Жученко, Олексій Кучкін, Артем Сазонов, Данило Згурський

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», просп. Перемоги 37, м. Київ, 03056, Україна

Анотація

Автоматичні системи контролю мобільними роботам досягають більшої ефективності за рахунок використання робастних алгоритмів навігації на основі оптичних датчиків, які формують тривимірну карту навколо об’єкту керування. Робота таких алгоритмів, зазвичай, призначена для: детектування ключових об’єктів навколишнього середовища; пошуку попередньо визначених об’єктів для релокації власного положення робота. Для вирішення проблеми детектування об’єктів з хмар точок існує багато різних підходів, але більшість з них мають високу обчислювальну складність. В цій роботі досліджено різні варіації методу консенсусу випадкової вибірки (RANSAC) для детектування об’єктів заданих математичною моделлю аналітичного виду. Для порівняння методів використані статистичні характеристики аналізу даних. Результати демонструють найбільш енергоефективний метод виявлення площин, який обробляє 60 кадрів RGB-D камери за секунду.

Ключові слова: консенсус випадкової вибірки; детектування площин; БПЛА.