

UDC 004.62:004.94(045)
DOI:10.18372/1990-5548.89.21488

¹Serhii Tsybulnyk,
²Vladyslav Shelemakha

SIMULATION OF THE RANSAC ALGORITHM TO DETERMINE OPTIMAL PARAMETERS

National Technical University of Ukraine "Ihor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine
E-mails: ¹tsybulnyk.serhii@edu.kpi.ua ORCID 0000-0002-4462-0936, ²anteriorard@gmail.com

Abstract—The work is devoted to the study and search for optimal parameters of the RANSAC algorithm (distance threshold, number of points and iterations) for segmenting the ground surface in order to increase the reliability of autonomous navigation in complex conditions. The open multisensor HOPS dataset containing point clouds of garden landscapes with a high level of geometric noise was used for modeling. The algorithm was implemented in Python 3.10 using the Open3D library. It was found that using the minimum 3 required number of points to construct a plane is the most effective strategy to achieve the target accuracy. It was found that at low iteration numbers, the random nature of the point selection dominates the useful signal, causing significant instability and high errors. To evaluate the modeling results, two relative error metrics were used, which are based on a comparison of four types of points: true positive, true negative, false positive and false negative. Specific parameter configurations were identified that ensure the stability of the results even in the presence of a significant number of outliers in the input data. The work allowed us to determine the structure of the RANSAC parameter space, delimiting sensitive local optima and robust stable regions. The formulated conclusions create a methodological basis for narrowing the range of parameters in future experiments and increasing the autonomy of unmanned systems in complex unstructured environments.

Keywords—RANSAC, geometric segmentation, LiDAR, autonomous navigation, parameter optimization.

I. INTRODUCTION

The development of autonomous mobile systems over the past decade has become one of the most dynamic areas in robotics [1], driven by the commercialization of unmanned ground (UGV) and aerial (UAV) vehicles for a wide range of applications: from precision agriculture and forest monitoring to industrial facility inspection and rescue operations. A fundamental prerequisite for the successful functioning of such systems is reliable navigation [2] – [4], which allows robots to perform multifunctional tasks under autonomous control, increasing the efficiency and quality of operations while simultaneously reducing labor costs. However, the complexity of modern operating environments – such as, for example, enclosed gardens, dense forests, multi-level poultry farms, or narrow underground tunnels – poses serious challenges for perception and positioning systems [3], [5] – [7].

Traditional navigation methods have long been based on global navigation satellite systems (GNSS), which can provide centimeter-level accuracy when using RTK (Real-Time Kinematic) technology [2], [8], [9]. However, in real operating conditions, GNSS demonstrates significant instability [2], [6], [10], [11]: signals are blocked by dense tree canopy, multi-beam reflection from tall buildings (the "urban canyon"

effect), or completely disappear in buildings and underground networks. This leads to errors that can reach several meters in the horizontal plane and tens of meters in the vertical plane [6], which is unacceptable for accurate drone navigation. In such GNSS-negative environments, there is an urgent need to use active localization methods that rely on onboard sensors to build maps and determine the current state of the robot.

Among existing sensor technologies, LiDAR (Light Detection and Ranging) technology occupies a leading position [3], [12], [13] due to its ability to create detailed 3D representations of the surrounding space regardless of lighting conditions, which makes it more effective compared to visual cameras that are sensitive to changes in light, fog, and atmospheric effects. LiDAR provides highly accurate distance measurements and allows building dense point clouds, which are the basis for simultaneous localization and mapping (SLAM) algorithms. However, the efficiency of SLAM and the accuracy of the trajectory directly depend on the quality of point cloud processing, where one of the most critical and at the same time complex stages is ground segmentation [14], [15].

The problem of geometric ground segmentation is the need to accurately extract a reference surface from an array of "raw" data to ensure stable

navigation. In loosely structured environments, such as agricultural fields or forests, the ground is rarely a perfectly flat plane – it is characterized by the presence of bumps, depressions, slopes, and irregularities. When a UGV moves over such terrain, it experiences significant oscillations along the roll and pitch axes, which leads to distortion of the LiDAR scan [2], [16], [17]: laser beams begin to capture a large number of soil points or, conversely, tree crowns at unpredictable angles. Without an effective segmentation algorithm, these "noise" points may be mistakenly identified as obstacles, causing the drone to stop or plan its path incorrectly.

Moreover, the presence of dense vegetation, weeds, or fallen leaves significantly complicates the process of differentiation of ground and objects [18], [19]. Traditional segmentation methods based on fixed height thresholds or simple Euclidean clustering often exhibit low adaptability: they may perceive tall grass as part of the ground or, conversely, ignore small obstacles on complex terrain. This leads to so-called "geometric ambiguities" when support posts in gardens or tree trunks are mixed with ground points, degrading the accuracy of constructed navigation lines [19]. Inaccurate segmentation is the root cause of accumulated drift in odometry algorithms [3], [10], [12], as erroneous scan matching leads to distortion of the constructed map and loss of robot localization in the long term.

Given algorithmic issues, modern solutions can often face a choice between accuracy and computational speed. Processing huge amounts of 3D LiDAR data in real time requires significant resources, which is critical for mobile platforms with limited battery and processor power. Algorithms based on deep learning (e.g., PointNet++ or RangeNet++) demonstrate [14], [20] high segmentation quality, but often require specialized graphics accelerators (GPUs) and large annotated datasets that are difficult to obtain for specific conditions. On the other hand, classical geometric methods, such as RANSAC for ground plane approximation, may be faster, but they are vulnerable to outliers and a large number of dynamic objects in the frame [5], [8] – people, animals, or other equipment moving in the navigation area. The impact of points from moving objects on a segment of the ground surface creates "ghosts" on odometry maps, which makes accurate loop closure impossible and destabilizes the entire optical navigation process [36].

In conditions where the ground remains the only stable geometric constant, and its precise definition allows imposing additional constraints on the

vertical axis (Z-axis), significantly reducing the positioning error, it is important to find a solution that will provide high segmentation accuracy and low hardware resource consumption to ensure the possibility of use on UGVs and UAVs.

II. MATERIALS AND METHODS

In the context of developing autonomous navigation systems for UGV and UAV based on LiDAR, point cloud segmentation methods act as a fundamental tool for interpreting the environment. The main task of segmentation [20] is to divide raw data into meaningful categories, such as the supporting surface (ground), obstacles, vegetation, or specific structural elements of the environment (tree trunks, supports, etc.). The effectiveness of SLAM navigation algorithms directly depends on the accuracy of this stage, as errors in point classification lead to accumulated odometry drift and the creation of incorrect patency maps. Among the wide range of approaches, geometric approximation methods occupy a special place, where the RANSAC (Random Sample Consensus) algorithm plays a central role [8], [21].

The RANSAC algorithm has proven itself as one of the most reliable [5], [22] methods for modeling geometric primitives in environments with high noise levels and a significant number of outliers. Unlike the classical least squares method, which attempts to minimize the total error for all points and is therefore extremely sensitive to outliers, RANSAC uses an iterative strategy of randomly selecting minimal subsets of the data to build model hypotheses.

In ground segmentation problems, this is usually [14] implemented through plane fitting, which is described by the equation $Ax + By + Cz + D = 0$. The process begins by randomly selecting three points that uniquely define a plane in 3D space. In this case, a discrete approximation of the three-dimensional space occurs in the form of a point cloud $P = \{p_1, p_2, \dots, p_n\}$, where each point p_i is determined by a vector of Cartesian coordinates (x, y, z) and an additional parameter of the intensity of the laser pulse reflection. This approach allows you to create accurate digital models of the environment, where the signal intensity is a critically important feature for differentiating individual objects.

After constructing the candidate model, the algorithm calculates the orthogonal distance from each point of the cloud to this plane using the formula [23]

$$d_i = \frac{|Ax_n + By_n + Cz_n + D|}{\sqrt{A^2 + B^2 + C^2}},$$

where d_i is the orthogonal (shortest) distance from a specific cloud point to the approximated plane (in the RANSAC algorithm, this value is compared with a threshold value to determine whether the point belongs to the ground surface (inlayer) or is an obstacle (outlayer), i is the index corresponding to the serial number of the calculated distance for each point in the current scan frame, A, B, C, D are parameters of the plane equation $Ax + By + Cz + D = 0$, which are selected by the RANSAC algorithm based on randomly selected points (vector (A, B, C) geometrically represents the normal to the plane), x_n, y_n, z_n are Cartesian coordinates of a specific point in the three-dimensional space of a LiDAR point cloud, n is the index of a point in a data array (point clouds).

In the context of ground segmentation using LiDAR, this formula determines how much each individual point cloud deviates from a mathematically constructed plane model. Points whose distance does not exceed a given threshold (for example, 0.05 m) are classified as inliers – that is, points belonging to the ground surface, and all other points are considered outliers. The optimal [22] plane model is considered to be that which maximizes the number of inlayers.

However, applying standard RANSAC in unstructured environments such as gardens or forests faces the problem of geometric ambiguity [16], [17]. To improve navigation accuracy, modern research proposes the introduction of additional restrictions on plane normal vectors [14]. In particular, when modeling the surface, the angle of inclination of the LiDAR relative to the horizon is taken into account: if the angle between the normal of the found plane and the vertical axis (0, 0, 1) exceeds a certain threshold, such a hypothesis is rejected as physically implausible. This avoids erroneous segmentation of vertical objects (e.g., walls or thick grass) as parts of the ground surface, which is critically important for the stable operation of LIO (LiDAR-Inertial Odometry) algorithms [9].

In addition to ground plane segmentation, RANSAC is widely used to form navigation lines in orchard rows. In this case, the algorithm approximates the rows of plants as straight lines $y = ax + b$ in a 2D projection [22]. To ensure autonomous robot movement, a combination of RANSAC with ROI (Region of Interest) pre-filtering is often used. For example, using Pass-Through filters [8], a working area around the chassis is selected, after which RANSAC iteratively identifies the left and right lines of the row, ignoring noise from weeds or dust. Moreover, custom

variations are being developed, such as Multi-line RANSAC [5], which are capable of simulating several parallel structures simultaneously (e.g., tunnel walls or rows in complex vineyards), which allows the device to be localized even in conditions of partial lack of data or sensor overlap.

The relevance of using RANSAC and clustering methods is also reinforced by the problem of LiDAR degradation in specific scenarios, such as "narrow corridors" or open fields. As mentioned above, in such conditions, the geometric features of the environment can be monotonous, which leads to orientation errors. Using RANSAC to identify the ground plane allows for tight constraints on the vertical orientation (Z-axis) and roll and pitch angles, which is critical to preventing vehicle drift. In addition, the intensity of the reflected signal can be integrated into the segmentation process: RANSAC can validate ground points not only by coordinates, but also by typical values of soil or asphalt reflectivity, which allows for additional filtering of dynamic objects (e.g., people in dark clothing) [9], [24].

It is also worth adding that modeling segmentation algorithms is an iterative process, where the choice of method depends on the balance between computational resources and reliability requirements. RANSAC remains the gold standard due to its ability to work with «dirty» data, where the noise fraction can reach 50% or more [5]. Combined with modern methods of statistical outlier removal and voxel grid to reduce data volume, these algorithms allow for the creation of high-precision navigation systems capable of ensuring the safety and autonomy of UGVs and UAVs in the most difficult operating conditions [12], [15]. That is why in this study, the RANSAC algorithm was chosen as the basis for geometric segmentation of the ground surface.

Determining the optimal RANSAC parameters, in particular the distance threshold and the number of iterations, is critically important to achieve a balance between surface modeling accuracy and minimal computational resource consumption. Incorrectly setting these values can lead to misclassification of obstacles as parts of the ground, causing accumulation of odometry errors and increasing the risk of a drone collision. Because RANSAC is stochastic in nature, careful parameter calibration helps minimize the method's inherent uncertainty and ensure stable results even when working with sparse or noisy point clouds. For this reason, the purpose of this work is to investigate and find the optimal parameters of the RANSAC algorithm for segmenting the ground surface in order

to increase the accuracy and reliability of navigation of unmanned ground and air systems in complex unstructured environments.

To verify the simulated algorithm based on the RANSAC method and assess its ability to adapt to real terrain conditions, the open multisensor HOPS dataset was used [25], [26]. The choice of this dataset, generated at the Klein-Altendorf research station (Bonn, Germany), is due to its high complexity and representativeness for autonomous navigation scenarios in intensive horticulture. Of particular value for modeling segmentation parameters is the hybrid nature of HOPS, which combines terrestrial laser scanning (TLS), mobile laser scanning (MLS), and photogrammetric reconstruction (SfM) data. The presence of point clouds (Fig. 1) with significantly different densities and levels of geometric noise allows for stress testing of the RANSAC algorithm when fitting the reference plane. In particular, the use of MLS and SfM from airborne platforms creates conditions close to UGV and UAV operational tasks, where the accuracy of ground allocation directly affects odometry stability and obstacle identification.

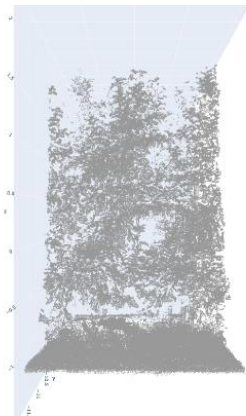


Fig. 1. Example of a point cloud from the HOPS dataset

The structure of the selected dataset contains detailed annotations for complex biological objects, which allows us to test the effectiveness of the proposed method in differentiating points on the ground surface from irrelevant structures – tree trunks, trellis wires, and support elements. This provides the ability to fine-tune the distance thresholds d_i , minimizing the risk of erroneous inclusion of garden infrastructure objects in the ground model, which is critical for building reliable navigation lines in unstructured agricultural environments.

The Python 3.10 programming language was used to develop and simulate the RANSAC algorithm, which is the de facto industry standard for computer vision and spatial data analysis tasks due to its

flexibility and compatibility with modern on-board computing platforms. The main software tool for the study is the Open3D library, a highly efficient open-source solution that provides support for complex data structures for processing 3D point clouds.

Data exchange was carried out through an object of the `open3d.geometry.PointCloud` class, which integrates with the NumPy library to perform vectorized coordinate normalization and preliminary noise filtering before activating the main segmentation stage. The procedure for geometric selection of the reference surface was implemented using the `segment_plane()` method, which automates the cyclic execution of the following stages.

1) From the initial array of points, the library's internal mechanisms randomly select N points (the mathematically necessary and sufficient minimum for building a model of the ground is $N = 3$). The input argument *ransac_n* corresponds to this number.

2) For each point p_i , the orthogonal distance to the approximated plane d_i is calculated. The point is identified as an "inlayer" (belonging to the ground) if this value does not exceed the *distance_threshold* (which is the second argument of the method).

3) The overall stability and accuracy of the fitting is determined by the number of iterations *num_iterations* (the third argument of the method), which, at an optimal value, allows the algorithm to achieve a stable result even in the presence of a high level of geometric noise and outliers in the "raw" LiDAR data.

Given the three input parameters, the search and evaluation of the optimal set is complicated by the number of their possible combinations. Therefore, the key methodological task was to choose an error function that correctly reflects not only the difference in the number of detected ground points in the point cloud, but also the degree of coincidence between the set of true and predicted points. For this, two complementary metrics were simultaneously used during the analysis of the results – E_{count} and E_{set} .

If we denote by TP (true positive) the number of points that the algorithm correctly assigned to the "ground" class, by FP (false positive) the number of points that were erroneously assigned to ground, and by FN (false negative) the number of true ground points that the algorithm did not detect, then the real number of ground points is $TP + FN$, and the number of points predicted as ground is $TP + FP$.

So, the E_{count} metric characterizes the relative error in the number of ground points. Its meaning is to estimate how much the total volume of the predicted ground differs from the true volume. In the transformed form, the metric is calculated through the difference between FP and FN and has the form:

$$E_{\text{count}} = \frac{|FP - FN|}{(TP + FN)}.$$

The E_{set} metric, on the other hand, is not focused on the balance of quantities, but on the quality of the reproduction of a set of points. It is determined through the Intersection over Union (IoU) index and has the form:

$$E_{\text{set}} = 1 - \text{IoU} = 1 - \frac{TP}{(TP + FP + FN)}.$$

Unlike E_{count} , this metric simultaneously responds to both excess points and missed points, preventing their mutual compensation. Thus, within the framework of this study, E_{set} is considered as the main quality metric, and E_{count} as an additional control characteristic. If two modes give close E_{set} values, then E_{count} allows us to understand whether this result is achieved due to an asymmetric balance between FP and FN. In turn, the simultaneous analysis of E_{set} , E_{count} and execution time allows us to move from a formal search for the minimum to a reasoned choice of practically feasible modes.

The main goal of analyzing the simulation results was not simply to find an isolated minimum error, but to identify such parameter configurations that simultaneously ensure: a) repeatability and stability of results between runs; b) monotonic or quasi-monotonic decrease in relative error with increasing number of iterations; c) proper quality according to the value of the main metric.

The use of this approach is explained by the stochastic nature of the RANSAC algorithm. Since the selection of subsets during hypothesis building has a random component, even with the same parameters, the algorithm can produce slightly different results. This means that the estimation of parameters from a single run of the algorithm can be unstable. Therefore, the basic unit of analysis in this work is not a single run, but a set of five runs on the same combination of parameters.

III. RESULTS AND DISCUSSION

The total number of selected scenes in the dataset is 82, but to narrow the range of parameters, it was decided to conduct the simulation on a smaller set of point clouds. Therefore, the initial set of simulation results contained 3015 rows of observations for one randomly selected scene, corresponding to combinations of three controlled parameters. The *distance_threshold* parameter took 201 values in the range from 0.001 m to 1,000 m, *ransac_n* – three values {3, 5, 7}, and *num_iterations* – five values {10, 50, 100, 250, 500}. Therefore, the total amount

of data for analysis was 15075 records – 5 runs of 3015 rows of observations to check the stability of the obtained results. In addition to the three control parameters, each record contained values of accompanying metrics for finding optimal ranges.

The formulation of the parameter selection problem in this case had a two-component nature. The first component was qualitative: it is necessary to ensure an error of no more than 2% as an optimal result or no more than 5% as acceptable. The second component was structural: we need not a single «best» point, but a parameter zone in which an increase in the number of iterations will not lead to chaotic jumps in the error, but will reproduce a physically and algorithmically plausible picture of error reduction. This means that the quality criterion was applied not only to the endpoint, but to the entire iteration window under study.

The simulation results, which are given in Table I, allow us to draw a fundamental conclusion regarding the iteration window. Although the full iteration grid 10–50–100–250–500 contained 50 monotonic pairs, none of them achieved stability and an acceptable quality level of $\leq 5\%$. This means that the inclusion of the smallest values of the number of iterations in the confirmatory scheme of the experiment significantly complicates the fulfillment of three requirements simultaneously – monotonicity, stability and low error. In contrast, the window 100–250–500 gave the largest number of monotonic-stable results (71) and the only optimal configuration in which all three error values are below 2%.

It was separately checked what contribution the *ransac_n* parameter makes to the formation of admissible modes. It turned out that all modes that remained in the set of candidates after imposing a strict quality criterion belong precisely to $N = 3$ (three points are enough to construct a plane). For $N = 5$ and $N = 7$, monotonic or stable trajectories were found, but they did not ensure simultaneous fulfillment of the error limit. Therefore, from the position of the target metric, it is $N = 3$ that should be considered as the base value for subsequent series of simulations.

After applying all the filters, four sets of parameters were identified (Fig. 2), which deserve separate consideration. One of them is optimal, i.e. it provides a relative error of no more than 2% over the entire iteration window of 100–250–500, and three are acceptable, i.e. those where all points do not exceed 5%. All these modes belong to *ransac_n* = 3, which further confirms the appropriateness of choosing this parameter as the base one.

The best mode is undoubtedly the configuration *distance_threshold* = 0.165 m; *ransac_n* = 3;

$num_iterations = 100-250-500$. The average error in this window is 1.570%, and the final one is 1.534%, i.e. all points are confidently placed in the optimal zone $\leq 2\%$. At the same time, the maximum deviation from the average error value over five runs within the window does not exceed 12.3%, which is

a completely acceptable result. It is also important that this mode is not "fragile": the transition from 100 to 250 and 500 iterations does not destroy either the stability or the shape of the trajectory (it is monotonic).

TABLE I. NUMBER OF PARAMETRIC PAIRS (DISTANCE_THRESHOLD, RANSAC_N) SATISFYING THE CRITERIA

Iterations window	Monotone pairs	Monotonous and stable
10–50–100	109	9
10–50–100–250	75	6
10–50–100–250–500	50	6
50–100–250	126	52
50–100–250–500	62	16
100–250–500	126	71

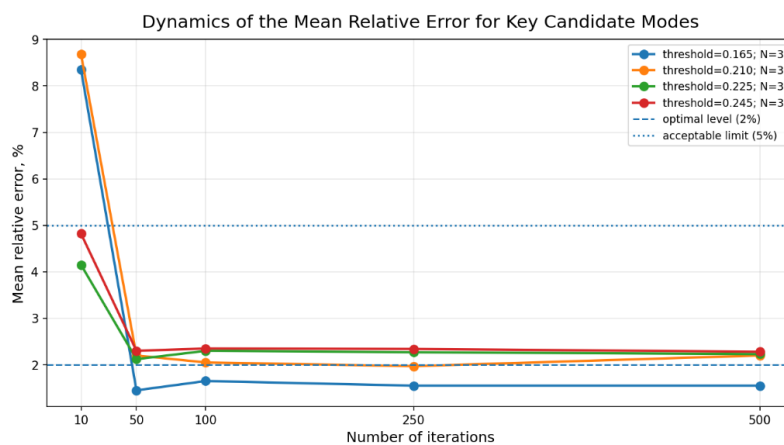


Fig. 2. Dynamics of the mean relative error

Figure 2 clearly demonstrates why the configuration $distance_threshold = 0.165$ m; $N = 3$ turned out to be the strongest. On the full iteration grid, it shows a sharp decrease in the error from 10 to 50 iterations, after which it passes into a low-level zone of about 1.5–1.6%. At the same time, at the point of 100 iterations, a slight increase in the average error is observed compared to 50 iterations, which is why the full window 50–100–250–500 cannot be considered monotonic. This is the reason for choosing the subwindow 100–250–500 as the main one. In other words, the algorithm behaves best not when the number of iterations increases unconditionally from minimum to maximum, but when the least informative and noisiest points of the initial stage are discarded.

The interpretation of the mode with 10 iterations is also important. For the basic configuration $distance_threshold = 0.165$ m; $N = 3$, the average error at 10 iterations is 8.372%, but the maximum deviation from the average reaches 345.625%. This means that with such a small computational resource, the stochastic component dominates the useful signal. Therefore, including the value 10 in the simulation confirmation window is

inappropriate, even if for some other configurations it sometimes gives acceptable results. From a practical point of view, 10 iterations can be considered only as a rough run or as a technical check of operability, but not as a component of the main evaluation mode.

Table II allows us to detail the behavior of the main recommended mode. At 50 iterations, a very low level of average error of 1.455% is already achieved, but the transition to 100 iterations is accompanied by a small increase to 1.637%. Further, a repeated decrease to 1.539% and 1.534% is observed for 250 and 500 iterations, respectively. That is, in the interval 100–500, a correct non-increasing trajectory is achieved, and the decrease after 250 iterations is already very small. This means that the point of 250 iterations is actually close to the saturation mode. Given the execution time, which increases from 1.239 s at 100 iterations to 1.617 s at 500, it is reasonable to consider 250 iterations as a basic compromise.

Based on the analysis performed, the main recommendation for further simulations should be considered a mode with parameters Distance threshold = 0.165 and Ransac $N = 3$, and it is

advisable to consider the number of iterations not as a separate number, but as a working window of 100–250–500. This approach is fully consistent with the requirement to take into account non-isolated values, namely the monotonic behavior of the error with the

other two parameters fixed. Within this window, all three points simultaneously satisfy the quality and stability criteria, and the deviations between sheets remain in a safe corridor.

TABLE II. NUMBER OF PARAMETRIC PAIRS (DISTANCE_THRESHOLD, RANSAC_N) SATISFYING THE CRITERIA

Number of iterations	Average error, %	Minimum by runs, %	Maximum by runs, %	Max. deviation from the mean, %	Average time, s
10	8.372	0.887	37.308	345.625	0.281
50	1.455	1.314	1.601	10.038	0.772
100	1.637	1.526	1.697	6.68	1.239
250	1.539	1.349	1.708	12.324	1.518
500	1.534	1.389	1.719	12.051	1.617

From a methodological point of view, it is advisable to recommend a further procedure for confirming the results. First, for any new set of input data, it is worth keeping at least five repeated runs, since it is on this number that the criterion for deviation from the mean was built. Second, when expanding or refining the grid of the Distance threshold parameter, it is advisable to perform search condensation in the vicinity of 0.150–0.200, since it is this region that contains the optimal mode and stable alternatives close to it.

It is also worth noting that the obtained recommendations are not a universal property of the algorithm for any data sets, but are statistically justified for the analyzed array of experiments. However, it is this form of recommendations – through the parameter zone and the permissible iteration window – that has an advantage over the fixed point, since it is better transferred to new series of simulations.

IV. CONCLUSIONS

For further serial simulations it is advisable to recommend a basic operating mode in the form of a corridor $distance_threshold = 0.150\text{--}0.200$ m, $ransac_n = 3$, $num_iterations = 250$ or 500 . If the task is to find the most portable single setting, it is advisable to start from the point $distance_threshold = 0.165$ m, $N = 3$, 250 iterations.

The given results allow us to significantly narrow the range of parameters in further numerical experiments, focusing on those $distance_threshold$ ranges and $ransac_n / num_iterations$ modes that have already demonstrated sufficient quality and repeatability on a large corpus of scenes. This, in turn, creates the prerequisites for building more compact and more representative experimental plans.

REFERENCES

[1] D. V. Nam et al., “Fusion consistency for industrial robot navigation: An integrated SLAM framework

with multiple2D LiDAR-visual-inertial sensors,” *Computers and Electrical Engineering*, vol. 120, 109607, 2024. <https://doi.org/10.1016/j.compeleceng.2024.109607>

- [2] Q. Li and H. Zhu, “Performance evaluation of 2D LiDAR SLAM algorithms in simulated orchard environments,” *Computers and Electronics in Agriculture*, vol. 221, 108994, 2024. <https://doi.org/10.1016/j.compag.2024.108994>
- [3] J. Jiang, T. Zhang, and K. Li, “LiDAR-based 3D SLAM for autonomous navigation in stacked cage farming houses: An evaluation,” *Computers and Electronics in Agriculture*, vol. 230, 109885, 2025. <https://doi.org/10.1016/j.compag.2024.109885>
- [4] J. Xiao et al., “An efficient, scalable, and high-precision multifunctional intelligent navigation system for agricultural machinery,” *Computers and Electronics in Agriculture*, vol. 242, 111336, 2026. <https://doi.org/10.1016/j.compag.2025.111336>
- [5] V. Androulakis et al., “Navigation system for a semi-autonomous shuttle car in room and pillar coal mines based on 2D LiDAR scanners,” *Tunnelling and Underground Space Technology incorporating Trenchless Technology Research*, vol. 117, 104149, 2021. <https://doi.org/10.1016/j.tust.2021.104149>
- [6] X. Zhang et al., “A dual-algorithm fusion positioning method based on global navigation satellite system signal quality assessment mode,” *Engineering Applications of Artificial Intelligence*, vol. 175, 114443, 2026. <https://doi.org/10.1016/j.engappai.2026.114433>
- [7] K. Usu et al., “Global self-localization and navigation using 3D-LiDAR for headland turning in vineyards,” *Smart Agricultural Technology*, vol. 12, 101296, 2025. <https://doi.org/10.1016/j.atech.2025.101296>
- [8] W. Liu et al., “Overall integrated navigation based on satellite and LiDAR in the standardized tall spindle apple orchards,” *Computers and Electronics in Agriculture*, vol. 216, 108489, 2024. <https://doi.org/10.1016/j.compag.2023.108489>

- [9] J. Cheng et al., "A tightly coupled integration of GNSS/IMU/LiDAR with parameterized semantic line and plane features to improve pose accuracy in complex environments," *Measurement*, vol. 247, 116843, 2025. <https://doi.org/10.1016/j.measurement.2025.116843>
- [10] J. Liu et al., "Tightly coupled GNSS RTK-LiDAR-inertial system for consistent urban navigation," *Measurement*, vol. 253, 117671, 2025. <https://doi.org/10.1016/j.measurement.2025.117671>
- [11] H. Tan et al., "A novel fusion positioning navigation system for greenhouse strawberry spraying robot using LiDAR and ultrasonic tags," *Agriculture Communications*, vol. 3, 100087, 2025. <https://doi.org/10.1016/j.agrcom.2025.100087>
- [12] S. Hammia et al., "Chapter 10 - A comprehensive review of LiDAR-SLAM for hardware implementation for autonomous vehicles," *Embedded Systems in Automotive Applications*, Elsevier, pp. 257–283, 2026. <https://doi.org/10.1016/B978-0-443-33861-8.00012-4>
- [13] A. H. Ahmed and H. Tomán, "A hybrid framework for UAV obstacle avoidance integrating reactive sensing, LiDAR planning, and deep reinforcement learning in real time," *Results in Engineering*, vol. 30, 110150, 2026. <https://doi.org/10.1016/j.rineng.2026.110150>
- [14] C. Ban et al., "Fusion of monocular camera and 3D LiDAR data for navigation line extraction under corn canopy," *Computers and Electronics in Agriculture*, vol. 232, 110124, 2025. <https://doi.org/10.1016/j.compag.2025.110124>
- [15] Z. Huang et al., "AELVI-SLAM: LiDAR-visual-inertial SLAM for autonomous exploration in multiple agricultural scenarios," *Artificial Intelligence in Agriculture*, vol. 16, pp. 940–957, 2026. <https://doi.org/10.1016/j.aiia.2026.03.013>
- [16] C. Ban et al., "A Camera-LiDAR-IMU fusion method for real-time extraction of navigation line between maize field rows," *Computers and Electronics in Agriculture*, vol. 223, 109114, 2024. <https://doi.org/10.1016/j.compag.2024.109114>
- [17] H. Teng et al., "Adaptive LiDAR odometry and mapping for autonomous agricultural mobile robots in unmanned farms," *Computers and Electronics in Agriculture*, vol. 232, 110023, 2025. <https://doi.org/10.1016/j.compag.2025.110023>
- [18] Y. Tsunoda, et al., "Power in numbers: Primitive algorithm for swarm robot navigation in unknown environments," *Robotics and Autonomous Systems*, vol. 198, 105329, 2026. <https://doi.org/10.1016/j.robot.2026.105329>
- [19] L.T. Rottok et al., "Development of a LiDAR-based framework for obstacle identification and mapping in orchard environments," *Biosystems Engineering*, vol. 257, 104245, 2025. <https://doi.org/10.1016/j.biosystemseng.2025.104245>
- [20] F. Li et al., "SD-SLAM: A semantic SLAM approach for dynamic scenes based on LiDAR point clouds," *Big Data Research*, vol. 36, 100463, 2024. <https://doi.org/10.1016/j.bdr.2024.100463>
- [21] B. P. Flavio et al., "LiDAR-only based navigation algorithm for an autonomous agricultural robot," *Computers and Electronics in Agriculture*, vol. 154, pp. 71–79, 2018. <https://doi.org/10.1016/j.compag.2018.08.034>
- [22] G. Betr` , S. Pascuzzi, and F. Paciolla, "A GNSS-free LiDAR-based navigation architecture for autonomous inter-row operation under sparse or absent vegetation conditions," *Smart Agricultural Technology*, vol. 14, 102106, 2026. <https://doi.org/10.1016/j.atech.2026.102106>
- [23] J. Pin et al., "Fast extraction of navigation line and crop position based on LiDAR for cabbage crops," *Artificial Intelligence in Agriculture*, vol. 15, pp. 686–695, 2025. <https://doi.org/10.1016/j.aiia.2025.03.007>
- [24] Z. Xu et al., "Intensity-enhanced LiDAR-inertial odometry with gradient flow sampling for degraded environments," *Measurement*, vol. 272, 121012, 2026. <https://doi.org/10.1016/j.measurement.2026.121012>
- [25] Gené-Mola et al., "Fuji-SfM dataset: a collection of annotated images and point clouds for Fuji apple detection and location using structure-from-motion photogrammetry," *Data in Brief*, vol. 30, 105591, 2020. <https://doi.org/10.1016/j.dib.2020.105591>
- [26] Gené-Mola J et al., "Fruit detection and 3D location using instance segmentation neural networks and structure-from-motion photogrammetry," *Computers and Electronics in Agriculture*, vol. 169, 105165, 2020. <https://doi.org/10.1016/j.compag.2019.105165>

Received: July 30, 2026

Accepted: August 19, 2026

Published: September 18, 2026

Tsybulnyk Serhii. ORCID 0000-0002-4462-0936. Candidate of Science (Engineering). Associate Professor. Faculty of Robotics and Instrumentation Engineering, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute," Kyiv, Ukraine.
Education: National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute," Kyiv, Ukraine.
Research interests: engineering analysis, automated software systems, data processing.
Publications: 43.
E-mail: tsybulnyk.serhii@edu.kpi.ua

Shelemakha Vladyslav. Postgraduate Student.

Faculty of Robotics and Instrumentation Engineering, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute," Kyiv, Ukraine.

Education: National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute," Kyiv, Ukraine.

Research interests: unmanned ground and aerial vehicles, data processing.

Publications: 2.

E-mail: anteri0r@gmail.com

С. О. Цибульник, В. В. Шелемаха. Моделювання алгоритму RANSAC для визначення оптимальних параметрів

Роботу присвячено дослідженню та пошуку оптимальних параметрів алгоритму RANSAC (дистанційного порогу, кількості точок та ітерацій) для сегментації поверхні землі з метою підвищення надійності автономної навігації в складних умовах. Для моделювання використано відкритий мультисенсорний набір даних HOPS, що містить хмари точок садових ландшафтів із високим рівнем геометричного шуму. Реалізація алгоритму виконана мовою Python 3.10 з використанням бібліотеки Open3D. Встановлено, що використання мінімально необхідної кількості точок для побудови площини є найбільш ефективною стратегією для досягнення цільової точності. Виявлено, що за низьких значень кількості ітерацій випадковий характер вибору точок домінує над корисним сигналом, спричиняючи значну нестабільність та високі похибки. Для оцінки результатів моделювання було використано дві метрики відносної похибки, які базуються на порівнянні чотирьох типів точок: істинно позитивних, істинно негативних, хибнопозитивних та хибнонегативних. Ідентифіковано специфічні конфігурації параметрів, які забезпечують стабільність результатів навіть у присутності значної кількості викидів у вхідних даних. Робота дозволила визначити структуру параметричного простору RANSAC, розмежувавши чутливі локальні оптимуми та робастні стабільні регіони. Сформульовані висновки створюють методологічну базу для звуження діапазону параметрів у майбутніх експериментах та підвищення автономності безпілотних систем у складних неструктурованих середовищах.

Ключові слова: RANSAC, геометрична сегментація, LiDAR, автономна навігація, оптимізація параметрів.

Цибульник Сергій Олексійович. ORCID 0000-0002-4462-0936. Кандидат технічних наук. Доцент.

Факультет робототехніки та приладобудування, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.

Освіта: Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.

Наукові інтереси: інженерний аналіз, автоматизовані програмні системи, обробка даних.

Кількість публікацій: 43.

E-mail: tsybulnyk.serhii@edu.kpi.ua

Шелемаха Владислав Валентинович. Аспірант.

Факультет робототехніки та приладобудування, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ, Україна.

Освіта: Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.

Наукові інтереси: безпілотні наземні та літальні апарати, обробка даних.

Кількість публікацій: 2.

E-mail: anteriorard@gmail.com