Validation of Robustness of SLAM Algorithms using Deep Learning Methods in Real Conditions

YURII RABESHKO, YURII TURBAL
Department of Applied Mathematics, Institute of Automation,
Cybernetics and Computer Engineering,
National University of Water and Environmental Engineering,
11 Soborna St, 33028, Rivne,
UKRAINE

Abstract: - As the level of modern technology development, namely autonomous robots, drones, robotics, etc., is high, the topic under study is highly relevant. Since the level of development of modern technologies, namely autonomous robots, drones, etc., is high, the topic under study is highly relevant. Due to the use of the Simultaneous Localisation and Mapping System (SLAM) in the industrial sector, ensuring and empirically verifying its robustness under challenging conditions is essential. The study aimed to evaluate and verify the reliability of the SLAM algorithm in real conditions. The following methods were used to conduct the study: deep learning methods and recurrent neural networks. ATE and RPE metrics were used to measure the accuracy of maps and trajectories. The study revealed a relatively high stability of the developed SLAM algorithm in changing lighting conditions and dynamic objects' presence. The ATE and RPE metrics were within acceptable limits. The study's scientific novelty and originality lie in considering the real conditions during the experiment, such as different lighting and dynamic objects, which were rarely considered in previous studies. The developed algorithm will be helpful for autonomous systems and in the context of the latest advanced technologies and robotics. A promising area for further research may be improving the SLAM algorithm for use in tough conditions.

Key-Words: - SLAM, CNN, RNN, deep learning, the robustness of algorithms, reliability, autonomous navigation.

Received: March 27, 2024. Revised: October 3, 2024. Accepted: November 5, 2024. Available online: December 3, 2024.

1 Introduction

SLAM (Simultaneous Localisation and Mapping) is a class of algorithms used in robotics and computer vision systems. Their functions are to determine a robot's or autonomous system's location in a certain environment and build a map of this environment.

The main components of SLAM include sensors, such as cameras, lidars, gyroscopes, and accelerometers, as well as algorithms for processing the data from these sensors. The robot or system first creates an initial estimated map and an estimate of its location and then gradually updates it by adjusting it with the data from the sensors.

SLAM algorithms have a wide range of applications in various fields, including mobile robots [1], [2]; Virtual and augmented reality, [3], [4]; environmental mapping, [5], [6]; delivery and logistics, [7], [8]; search and rescue, [9], [10]:

The development and improvement of SLAM algorithms are significant for society's technological development. This makes them particularly useful for a wide variety of applications. AR and VR use SLAM to interact with the real world, create and track objects, and provide highly realistic gaming experiences. SLAM autonomously navigates drones for various purposes, including delivery, aerial photography, and search and rescue operations. Autonomous vehicles use SLAM to safely navigate and locate themselves on the road, improving road safety and reducing accidents. In defense and security, SLAM can be used in military applications for autonomous missile and drone guidance and navigation in unknown areas.

Validation of the robustness of SLAM algorithms is also of great importance. Since these algorithms operate in different, sometimes harsh environments, their robustness is critical. SLAM errors can lead to dangerous situations in the above applications. Therefore, validation helps ensure the algorithms function correctly, including limited visibility, many entities on the map, changing lighting, etc. Increasing the robustness of SLAM algorithms is an important task to improve the reliability and safety of autonomous systems in the modern world.

Given the high demand and usefulness of SLAM algorithms in many application areas, **this study aims** to investigate the robustness of the developed SLAM algorithm by testing it in real-world conditions. To achieve this goal, the following tasks have been set and implemented:

1. Development of a SLAM algorithm using deep learning methods.
2. Testing in actual conditions, mainly conducting experiments with complications (emergencies, etc.).
3. Analysing the results according to the selected metrics.

2 Literature Review

Today, SLAM systems with deep learning are increasingly being used. They have significant advantages that help improve the accuracy and reliability of SLAM systems, namely:

- Increased localization accuracy;
- Improving the accuracy and information content of maps;
- improving the resistance of the SLAM system to various factors, such as variable illumination, glare, or interference;
- reducing sensor data processing time and increasing the SLAM system's response to changes in the environment, which is especially important for real-time applications;
- ability to work under challenging conditions (limited visibility).

Deep learning helps make SLAM systems more efficient, reliable, and flexible in different environments and applications. It opens new possibilities for autonomous systems and allows them to understand their environment more accurately and navigate it efficiently.

Therefore, there is a lot of research and development in this area. For example, authors, [11], proposed depth estimation algorithms for an image based on SLAM and CNN (convolutional neural networks) to compact and scale data and provide a real-time environment map suitable for real-time exploration. Authors, [12], proposed to combine the potential of feature descriptors based on deep learning with traditional VSLAM to increase the reliability of a conventional VSLAM system. Their results show an increase in the performance and reliability of the algorithm and resistance to sensory noise. The authors, [13], also proposed a real-time visual SLAM algorithm based on a deep learning method – the YOLOv5s convolutional neural network as a parallel semantic thread.

A significant segment is the development of SLAM for work in dynamic environments, as it opens up new opportunities for the development of autonomous systems and the growth of robotics in various fields while improving safety, efficiency, and quality of life. Therefore, quite a few publications are devoted to this area. For example, [6], describes a Dynamic-SLAM based on a convolutional neural network with an improved performance of 10%. A similar study was conducted, [14], which describes the SLAM algorithm with multi-target tracking SLAMMTT. Also, authors, [15], describe DDL-SLAM (Dynamic Deep Learning SLAM), a robust RGB-D SLAM system for dynamic scenarios, which adds dynamic object segmentation capabilities based on ORB-SLAM. Authors, [16], propose a new approach to visual Graph-SLAM for underwater robots; their model is also based on a Siamese convolutional network SCNN, which is designed to be easily trainable and compares pairs of underwater images, rejecting those that do not close the loop. The tests were conducted on semi-synthetic data. Deep learning methods are also used by, [17], in combination with LIDAR SLAM for people detection for autonomous indoor mapping in a populated environment. Authors, [18], propose a new dynamic RGB-D SLAM method, PLD-SLAM, based on point and linear functions for dynamic scenes. PLDSLAM contains a deep learning algorithm for segmenting semantic information and a K-Means clustering algorithm that considers the deep information to recognize dynamic features. Then, two consistency-checking methods are used to filter out the dynamic features. After a thorough analysis, it can be noted that PLD-SLAM performs better than conventional SLAM under dynamic conditions.

In, [19], GO-SLAM, a visual SLAM framework based on deep learning that performs real-time 3D reconstruction, is presented. It is based on pose estimation and online full-package adjustments that optimize the frame by using the learned global geometry of the complete history of the input frames. In addition, GO-SLAM is versatile and can work with monocular, stereo, and RGB-D input.

So, recently, advances in the development of SLAM algorithms have achieved significant results, especially with the use of deep learning methods, mostly convolutional networks. Nevertheless, there are still many problems related to ensuring the robustness of the developed algorithms and many ways to solve them. Therefore, this paper is devoted to a detailed validation of the robustness of the developed SLAM algorithm.

3 Methods and Materials

SLAM algorithm using deep learning:

1. Sensor information: the DJI MAVIC 3T drone collects visual data from its cameras, including the zoom and thermal cameras. It also uses vision and infrared sensors to detect obstacles.
2. Preliminary data processing: visual data is processed to identify critical points and image features. Data from other sensors are also processed to create a comprehensive set of environmental data.
3. Deep learning model (CNN): A deep convolutional neural network (CNN) trained on a large amount of data determines the depth of objects in images, which helps reconstruct a three-dimensional map of the environment.
4. Sequential analysis (RNN): A Recurrent Neural Network (RNN) combines information from different frames and follows the movement of objects and the localization of the drone. RNN helps to consider the temporal sequence and provides consistent localization and mapping.
5. Data fusion: Information from CNN and RNN is combined to create a comprehensive 3D map of the environment and build the drone's route. Based on new data, localization and positioning parameters are updated in real-time.
6. Validation and testing: The obtained map and traffic information are checked and validated based on known data from actual flights. The network and algorithm parameters are tuned for the best possible accuracy and reliability.

Parameters and hyperparameters for the convolutional neural network CNN:

Network architecture: deep convolutional network with ten convolutional layers and three fully connected layers.

Filter size: 3x3 for all convolutional layers.

Number of filters: 64 in the first two convolutional layers and 128 in the subsequent layers.

Activation function: ReLU for all layers.

Pulling parameters: MaxPooling with a size of 2x2 after each convolutional layer.

Loss function: Mean square error (MSE) for the loss function for depth reconstruction.

The learning rate starts at 0.001 and adapts during training, reducing it by 50% if necessary.

Parameters and hyperparameters for a recurrent neural network (RNN):

RNN architecture: LSTM layers for modeling time dependencies.

Number of layers: two LSTM layers for recurrent analysis.

The size of the hidden state: 256 units for each LSTM layer.

Activation function: hyperbolic tangent (tanh) of all LSTM layers.

Optimizer parameters: Adam with a learning rate of 0.001.

The internal parameters of SLAM are adjusted automatically during operation using auto-calibration methods and optimization techniques. This approach allows the SLAM algorithm parameters to be adapted to changing conditions and environments.

The block diagram of the described algorithm is shown in Figure 1 (Appendix).

3.1 Main Technical Characteristics

For the aim of our research, we used a DJI MAVIC 3T drone with the following characteristics:

Operational:

- power supply - battery, Li-pol type;
- battery capacity - 5000 mAh;
- the warranty period is 24 months;
- the parameters of the device are 347.5 x 283 x 107.7 mm;
- weight - 920 grams;
- the case material is plastic;
- temperature conditions of use - from -10 to +40°C.

Multimedia:

- internal memory - 8 GB;
- connection - Glonass, GPS;
- camera - 20 MP, CMOS 4/3;
- the maximum altitude is 6000 meters;
- the maximum climb speed is 68 km/h;
- the maximum permissible wind speed is 43.2 km/h;
- control - remote control;
- remote control frequency - 2.4 GHz, 5.8 GHz;
- flight time - up to 45 minutes;
- climbing speed - 28.8 km/h;
- the descent speed is 21.6 km/h.

Screen:

- 4K video recording resolution: 3840x2160, 30 fps, FHD: 1920x1080, 30 fps;
- The resolution is 8000x6000 pixels.

Additional features

- a complete software package;
- RTK module;
- mobile station D-RTK 2.

These indicators make the drone highly popular among users, as evidenced by their positive feedback.

3.2 Camera

The UAV has a powerful dual camera, a high-quality 4/3 CMOS sensor, and a reliable radiometric thermal sensor. These sensors provide high-quality thermal images to capture important information. The camera's three-axis stabilization produces 4K video at 30 frames per second. The x56 digital hybrid zoom provides maximum image detail.

3.3 Energy Supply

The quadcopter is equipped with a 5000 mAh lithium-polymer battery. The increased capacity ensures the device's autonomy. The battery has an increased level of safety, which reduces the risk of spontaneous combustion. Due to a slight voltage drop during discharge, the battery retains its performance for a long time without recharging. The battery's temperature range is from -20 to +50°C.

The minimal thickness of the battery contributes to the compactness and improved ergonomics of the device.

3.4 Speed, Altitude and Distance

The maximum rate of climb reaches 68 km/h, and the average rate is 28.8 km/h. The descent speed is within 21.6 km/h. The quadcopter can fly to an altitude of up to 6000 meters.

3.5 Thermal Imager

The quadcopter has a powerful thermal imaging camera with a vanadium oxide matrix. The thermal imager has a lens with an aperture of f/1.0 and a viewing angle of 61°. Automatic focus is triggered at a distance of 5 meters and is valid indefinitely. The sensitivity reaches ≤50 mK. Video resolution is 640 x 512 pixels at 60 fps. The wavelength of infrared radiation is from 8 to 14 µm. The temperature range of object recognition in the high gain mode varies from -20 to +150°C, and in the low gain mode, it ranges from 0 to 500°C. The measurement accuracy has an error of ± 2%.

Due to the complexity of the process of assessing the stability of the SLAM algorithm, it is important to take into account many factors. To do this, conducting an experiment in natural conditions and using objective values to assess the effectiveness is necessary. Therefore, the following aspects were taken into account to assess the effectiveness of the algorithm:

1. The ability to work in dynamic conditions (lighting, weather conditions, object movement, etc.). To do this, it is important to assess the algorithm's adaptability to changes and ensure that it continues to work reliably.
2. Ability to recognize dynamic objects (people, vehicles, etc.).
3. Long-term stability (determining how the algorithm can work over time). This is important to prevent the effect of error accumulation.
4. Ability to continue working in case of data loss (loss of sensors or cameras).
5. Check for internal errors.

3.6 Metrics for Evaluating the Results of Experiments

The following indicators were used to assess the accuracy and reliability of the SLAM algorithm:

1. Absolute trajectory error (ATE).

The following formula is used to calculate ATE:

$$ATE = \sqrt{(\sum_{i=1}^{n}(|p\_predicted_i - p\_ground\_truth_i|^2)/n}$$

Where:

p_predicted - position calculated by the SLAM algorithm;

p_ground_truth - known (ideal) position;

n - number of points to compare on the trajectory.

2. Relative position error (RPE).

The following formula is used to calculate RPE:

$$RPE = \sqrt{(\sum_{i=1}^{n}(|p\_predicted_i - p\_previous_i|^2)/n}$$

Where:

p_predicted - the current position calculated by the SLAM algorithm.

p_previous - the previous known position.

n - number of pairs of items to compare.

These metrics measure the accuracy and stability of a SLAM system's localization relative to known positions and trajectories. They help developers and researchers assess the quality and robustness of an algorithm, conduct a comparative analysis of different SLAM implementations, and improve algorithms to achieve better results in different conditions.

4 Results

To evaluate the robustness of the algorithm described in the previous section, we conducted experiments under the following conditions:

1. Daylight, excluding dynamic objects.
2. Night lighting, excluding dynamic objects.
3. Daytime lighting with dynamic object recognition.
4. Night lighting with dynamic object recognition.
5. Daytime lighting, presence of obstacles.

100, 500, and 1000 m runs were performed for each set of conditions, and each experiment was repeated five times. Errors of the ATE (m), RPE (m), and RMSE (m) results are shown in Appendix in Table 1, Table 2, Table 3, Table 4, Table 5 and also depicted in the histograms (Figure 2, Figure 3 and Figure 4).

According to Table 1 (Appendix), in daylight (sufficient) conditions and when dynamic objects are ignored, the developed algorithm shows satisfactory results, which deteriorate with increasing drone flight range. The range of error values over several tests changes insignificantly and increases almost proportionally to the increase in flight range.

According to Table 2 (Appendix), which describes the experiment in night (low) light conditions and when dynamic objects are ignored, the developed algorithm shows satisfactory results that deteriorate with increasing drone flight range, as in the case of daylight. Comparing the data in Appendix in Table 1 and Table 2, we can see a slight deterioration in the results when the lighting changes. This is within acceptable limits, given the scale of the system's complexity. The range of error values over several tests changes slightly and increases almost proportionally to the increase in flight range.

Table 3 (Appendix) shows the experiment results in daylight and with the additional task of recognizing dynamic objects. In these conditions, the algorithm under study shows approximately 1.5 times worse results than in Experiment 1, where dynamic objects were ignored. Nevertheless, the results remain satisfactory, and the range of error values changes only slightly over several trials. This indicates the reliability of the algorithm.

Table 4 (Appendix) shows the experiment's results under night lighting and with the additional task of recognizing dynamic objects. Comparing these error values with the values in Table 3 (Appendix), we conclude that the change in lighting does not critically affect the algorithm's performance when it’s necessary to recognize dynamic objects. Compared to the results of Experiment 2 (night lighting, ignoring dynamic objects), the results of this experiment deteriorate by an average factor of 1.5, as well as when comparing Experiments 1 and 3.

Table 5 (Appendix) shows the experiment results in daylight, ignoring dynamic objects but with communication hindrances. The algorithm continues to work in these conditions and demonstrates slightly higher errors than in the experiment under similar conditions but with reliable communication. At the same time, the error values obtained in Experiment 5 are better than those obtained in dynamic systems and static systems under night lighting.

Figure 2, Figure 3 and Figure 4 show the histograms of the mean values of ATE (m), RPE (m), and RMSE (m) errors for each experiment, respectively.

Figure 2 shows each experiment's histograms of the average trajectory error values. As indicated in this figure, the errors increase proportionally with the drone's flight range, which is quite natural. The highest error values are found in Experiment 4, which has the most challenging conditions and tasks.

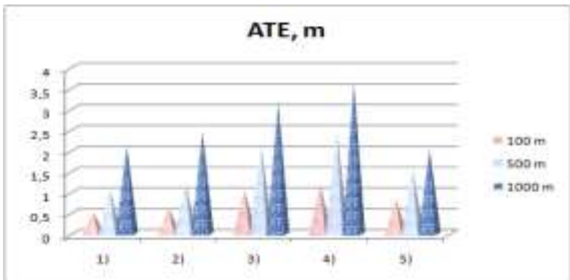

Fig. 2: Histograms of the mean values of the ATE error for each experiment.

Figure 3 shows histograms of the average position error values for each experiment. As in the case of trajectory error, the errors increase proportionally to the increase in the drone's flight range. Also, according to the results obtained, it can be argued that the position determination is entirely accurate, given that the highest RPE value of 0.182 m is found in Experiment 4 under the most challenging conditions, task, and most extended range.

Figure 4 illustrates the histograms of the mean values of each experiment's root mean square mapping error. As in the case of trajectory and position errors, the errors increase proportionately to the drone's flight range increase.

The results generally indicate that the developed SLAM algorithm is highly robust and adaptable to environmental conditions and problem formulation changes.

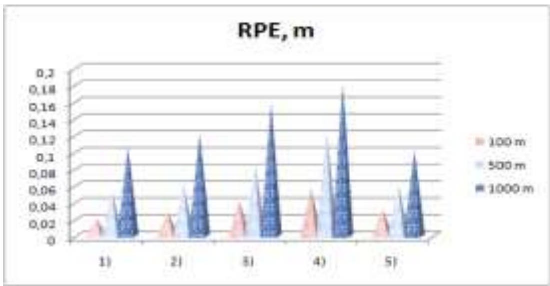

Fig. 3: Histograms of mean RPE error values for each experiment.

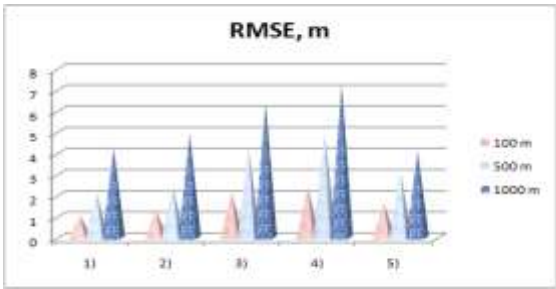

Fig. 4: Histograms of average RMSE error values for each experiment.

5 Discussion

This paper empirically verifies and confirms the robustness of the developed SLAM algorithm through a series of experiments conducted under actual conditions.

The described algorithm demonstrated satisfactory results when environmental conditions change (day and night lighting), interference occurs (with the presence of dynamic objects), and when the task becomes more complex, such as obstacles or loss of communication. Based on deep learning (using convolutional and recurrent neural networks), this algorithm is promising and competitive. For example, [10], conducted a similar study but focused on the low hardware cost and neglected achieving high accuracy. As noted: “Although the quality of the generated point clouds is still low and not comparable to photogrammetric reconstructions, the scene scale is correctly scaled”, [10]. In contrast, the algorithm developed in this study has a more complex structure and sophisticated topology of the neural networks used due to its much higher accuracy.

In colleagues' works, [11], [14], the convolutional network CNN is used in the presented SLAM algorithms using artificial neural networks of deep learning. Other convolutional networks were used, [13], [17]: YOLOv5s and SCNN, respectively. Instead, the algorithm described in this paper combines the work of CNN and RNN, which provides better performance and significantly increases the algorithm's reliability.

In, [19], the SLAM algorithm combines a deep learning neural network with a clustering method, which is more capacious and less reliable than the consolidation of CNN and RNN.

In addition, none of the reviewed works, [12], [15], paid sufficient attention to the data loss scenario (sensors or cameras). Therefore, this work is valuable because of the developed algorithm and the study of its robustness in emergencies.

For example, in, [20], an algorithm using semantic segmentation DeeplabV3+ was applied to recognize dynamic objects, but no experiments were conducted with changing lighting. Instead, the authors describe experiments with the presence of dynamic objects in both day and night lighting. After all, this is a very important aspect of operating drones and autonomous robots in real-world conditions. Therefore, the algorithm should be tested in all possible scenarios. This is the only way to prove the robustness of the developed algorithm.

A significant advantage of this study is that the algorithm was tested in real-world conditions rather than artificially simulated conditions, as in, [21].

The developed algorithm can be applied in several areas: autonomous drones, robotic environments with limited access, robots for exploring dangerous or hard-to-reach places, mobile robots in challenging environments, robots working in harsh conditions, such as during rescue operations, underwater exploration, etc.

6 Conclusions

Simultaneous localization and mapping (SLAM) algorithms are crucial to navigating mobile robots, drones and autonomous vehicles. However, their successful implementation in real-world environments requires research and verification of robustness. In this context, the results of the experiments, which include measurements of absolute trajectory error, relative position error, and mapping error, are significant.

The developed SLAM algorithm using deep learning (CNN, RNN) was tested in several different experiments. Considering the data presented in the tables, the best results are achieved in daytime flight conditions by ignoring dynamic objects at distances of 100 meters. However, the results obviously and uncritically deteriorate with increasing distance, in night conditions, or when dynamic objects are detected. This proves the robustness of the algorithm under study.

The developed SLAM algorithm can be used in various applications, such as autonomous navigation of drones and robots in demanding environments, virtual reality, augmented reality, and robotics for research and rescue operations.

Future research should focus on improving the robustness and reliability of SLAM algorithms in real-world environments. This may include developing new methods for filtering and compensating for sensor errors, adapting algorithms to changing conditions, and extending their application to a broader range of scenarios.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work the authors used ChatGPT in order to improve readability and language. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References:

[1] Singandhupe, A., and La, H. M. A review of slam techniques and security in autonomous driving. *2019 Third IEEE International Conference on Robotic Computing (IRC)*. IEEE, Naples, Italy, 2019. https://doi.org/10.1109/IRC.2019.00122.

[2] Nobis, F., Papanikolaou, O., Betz, J., Lienkamp, M. Persistent map saving for visual localization for autonomous vehicles: An orb-slam 2 extension. *arXiv*. https://doi.org/10.48550/arXiv.2005.07429

[3] Jinyu, L., Bangbang, Y., Danpeng, C., Nan, W., Guofeng, Z., Hujun, B. Survey and evaluation of monocular visual-inertial SLAM algorithms for augmented reality. *Virtual Reality & Intelligent Hardware*, Vol.1, No.4, 2019, pp. 386–410.

[4] Bettens, A. M., Morrell, B., Coen, M., McHenry, N., Wu, X., Gibbens, P., Chamitoff, G. UnrealNavigation: Simulation software for testing SLAM in virtual reality. *AIAA Scitech 2020 Forum*, Orlando, FL, USA, 2020. https://doi.org/10.2514/6.2020-1343.

[5] Shu, F., Lesur, P., Xie, Ya., Pagani, A., Stricker, D. SLAM in the field: An evaluation of monocular mapping and localization on challenging dynamic agricultural environment. *Proceedings of the IEEE/CVF winter conference on applications of computer vision. IEEE,* Waikoloa, HI, USA, 2021. https://doi.org/10.1109/WACV48630.2021.00180.

[6] Xiao, L., Wang, J., Qiu, X., Rong, Zh., Zou, X. Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment. *Robotics and Autonomous Systems*, Vol.117, 2019, pp. 1-16. https://doi.org/10.1016/j.robot.2019.03.012.

[7] Zhang, J., Zhang, X., Shen, X., Wu, J., Li, Yu. A Lidar SLAM based on Improved Particle Filter and Scan Matching for Unmanned Delivery Robot. *Journal of Physics: Conference Series (IOP Publishing).* Vol.2506, 2023, paper 012009, https://doi.org/10.1088/1742-6596/2506/1/012009.

[8] Wang, Wenmin, Zhao, W., Wang, X., Jin, Zh., Li, Yu., Runge, T. A low-cost simultaneous localization and mapping algorithm for last-mile indoor delivery. Proceedings from *2019 5th International Conference on Transportation Information and Safety (ICTIS)*. IEEE, Liverpool, UK, 2019. https://doi.org/10.1109/ICTIS.2019.8883749.

[9] Wang, H., Zhang, Ch., Song, Y., Pang, B., Zhang, G. Three-dimensional reconstruction based on visual SLAM of mobile robot in search and rescue disaster scenarios. *Robotica*, Vol.38, No.2, 2020, pp. 350-373. https://doi.org/10.1017/S0263574719000675.

[10] Yang, Zh., Nabila Naz, N., Pengcheng Liu, P., Nazmul Huda, M. Evaluation of SLAM algorithms for search and rescue applications. Annual Conference Towards Autonomous Robotic Systems. *Cham: Springer Nature* Switzerland, 2023, pp. 114-125.

[11] Steenbeek, A., Nex, F. CNN-based dense monocular visual SLAM for real-time UAV exploration in emergency conditions. *Drones*, Vol.6, No.3, 2022, paper 79. https://doi.org/10.3390/drones6030079.

[12] Bruno, H. M. S., Colombini, E. L. LIFT-SLAM: A deep-learning feature-based monocular visual SLAM method. *Neurocomputing*, Vol.455, 2021, pp. 97-110. https://doi.org/10.1016/j.neucom.2021.05.027.

[13] Su, P., Luo, S., Huang, X. Real-time dynamic SLAM algorithm based on deep learning. *IEEE Access*, Vol.10, 2022, pp. 87754-87766.

https://doi.org/10.1109/ACCESS.2022.3199350.

[14] Bahraini, M. S., Rad, A. B., Bozorg, M. SLAM in dynamic environments: A deep learning approach for moving object tracking using ML-RANSAC algorithm. *Sensors*, Vol.19, No.17, 2019, paper 3699. https://doi.org/10.3390/s19173699.

[15] Ai, Y., Rui, T., Lu, M., Fu, L., Liu, S., Wang, S. DDL-SLAM: A robust RGB-D SLAM in dynamic environments combined with Deep Learning. *IEEE Access*, Vol.8, 2020, pp. 162335–162342. https://doi.org/10.1109/ACCESS.2020.2991441.

[16] Burguera, A., Bonin-Font, F., Guerrero Font, E., Martorell Torres, A. Combining deep learning and robust estimation for outlier-resilient underwater visual graph SLAM. *Journal of Marine Science and Engineering*, Vol.10, No.4, 2022, paper 511. https://doi.org/10.3390/jmse10040511.

[17] Tiozzo Fasiolo, D., Maset, E., Scalera, L., Macaulay, S. O., Gasparetto, A., Fusiello, A. Combining LiDAR SLAM and deep learning-based people detection for autonomous indoor mapping in a crowded environment. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (pp. 447-452), Vol.43. XXIV ISPRS Congress, Nice, France, 2022, [Online]. https://pdfs.semanticscholar.org/6de8/180238803ae6442c4108d94d3343c33e40dc.pdf (Accessed Date: January 13, 2024).

[18] Zhang, Ch., Huang, T., Zhang, R., Yi, X. PLD-SLAM: A new RGB-D SLAM method with point and line features for indoor dynamic scene. *ISPRS International Journal of Geo-Information*, Vol.10, No.3, 2021, paper 163. https://doi.org/10.3390/ijgi10030163.

[19] Zhang, Y., Tosi, F., Mattoccia, S. Poggi, M. Go-slam: Global optimization for consistent 3D instant reconstruction. *Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 3727-3737)*, IEEE, Paris, France 2023. https://dx.doi.org/10.1109/ICCV51070.2023.00345.

[20] Zhang, X. Y., Rahman, A. H. A., Qamar, F. Semantic visual simultaneous localization and mapping (SLAM) using deep learning for dynamic scenes. *PeerJ Computer Science*, Vol.9, 2023, paper e1628. https://doi.org/10.7717/PEERJ-CS.1628.

[21] Wang, B., Cheng, X., Wang, J., Jiao, L. A Bidirectional scoring strategy-based transformation matrix estimation of dynamic factors in environmental sensing. *Remote Sensing*, Vol.16, No.4, 2024, paper 723. https://doi.org/10.3390/rs16040723.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from formulating the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare.

APPENDIX

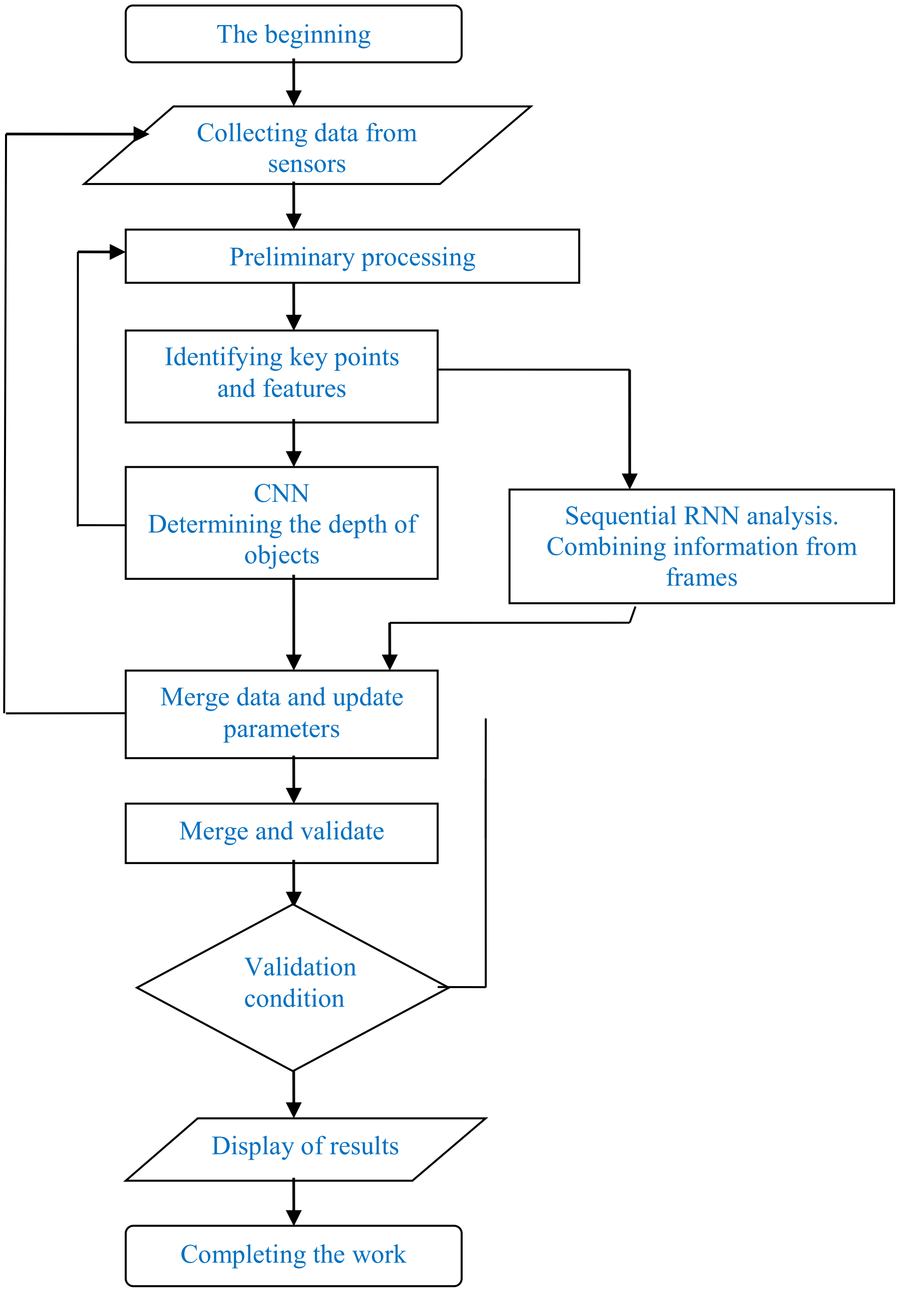

Fig. 1: Block diagram of the algorithm

Table 1. Errors of the SLAM algorithm in the first experiment

Conditions	1. Daylight, excluding dynamic objects								
Range	100			500			1000		
Error	ATE	RPE	RMSE	ATE	RPE	RMSE	ATE	RPE	RMSE
1	0,5	0,02	1	1	0,05	2	2	0,1	4
2	0,6	0,025	1,2	1,2	0,06	2,4	2,4	0,12	4,8
3	0,55	0,022	1,1	1,1	0,055	2,2	2,2	0,11	4,4
4	0,48	0,018	1,05	0,95	0,048	1,95	1,9	0,095	3,8
5	0,52	0,021	1,08	1,05	0,052	2,1	2,1	0,105	4,2
Average	0,53	0,0212	1,086	1,06	0,053	2,13	2,12	0,106	4,24
Span	0,12	0,007	0,2	0,25	0,012	0,45	0,5	0,025	1

Table 2. Errors of the SLAM algorithm in the second experiment

Conditions	2. Night lighting, excluding dynamic objects								
Range	100			500			1000		
Error	ATE	RPE	RMSE	ATE	RPE	RMSE	ATE	RPE	RMSE
1	0,6	0,03	1,2	1,2	0,06	2,4	2,4	0,12	4,8
2	0,7	0,035	1,4	1,4	0,07	2,8	2,8	0,14	5,6
3	0,65	0,028	1,3	1,3	0,065	2,6	2,6	0,13	5,22
4	0,58	0,025	1,25	1,1	0,055	2,2	2,2	0,11	4,4
5	0,62	0,027	1,28	1,15	0,057	2,3	2,3	0,115	4,6
Average	0,63	0,029	1,286	1,23	0,0614	2,46	2,46	0,123	4,924
Span	0,12	0,01	0,2	0,3	0,015	0,6	0,6	0,03	1,2

Table 3. Errors of the SLAM algorithm in the third experiment

Conditions	3. Daylight, with dynamic object recognition								
Range	100			500			1000		
Error	ATE	RPE	RMSE	ATE	RPE	RMSE	ATE	RPE	RMSE
1	1	0,04	2	2	0,08	4	3	0,15	6
2	1,2	0,05	2,4	2,4	0,096	4,8	3,6	0,18	7,2
3	1,1	0,045	2,2	2,2	0,088	4,4	3,3	0,165	6,6
4	0,95	0,038	1,95	1,9	0,076	3,8	2,85	0,142	5,7
5	1,05	0,042	2,1	2,1	0,084	4,2	3,15	0,157	6,3
Average	1,06	0,043	2,13	2,12	0,0848	4,24	3,18	0,1588	6,36
Span	0,25	0,012	0,45	0,5	0,02	1	0,75	0,038	1,5

Table 4. Errors of the SLAM algorithm in the fourth experiment

Conditions	4. Night lighting with dynamic object recognition								
Range	100			500			1000		
Error	ATE	RPE	RMSE	ATE	RPE	RMSE	ATE	RPE	RMSE
1	1,2	0,06	2,4	2,4	0,12	4,8	3,6	0,18	7,2
2	1,3	0,065	2,6	2,8	0,14	5,6	4	0,2	8
3	1,25	0,058	2,5	2,6	0,13	5,2	3,8	0,19	7,6
4	1,1	0,052	2,2	2,2	0,11	4,4	3,3	0,165	6,6
5	1,15	0,055	2,3	2,3	0,115	4,6	3,5	0,175	7
Average	1,2	0,058	2,4	2,46	0,123	4,92	3,64	0,182	7,28
Span	0,2	0,013	0,4	0,6	0,03	1,2	0,7	0,035	1,4

Table 5. Errors of the SLAM algorithm in the fifth experiment

Conditions	5. Daylight, with obstacles								
Range	100			500			1000		
Error	ATE	RPE	RMSE	ATE	RPE	RMSE	ATE	RPE	RMSE
1	0,8	0,03	1,6	1,5	0,06	3	2	0,1	4
2	0,9	0,035	1,8	1,7	0,068	3,4	2,2	0,11	4,4
3	0,85	0,032	1,7	1,6	0,064	3,2	2,1	0,105	4,2
4	0,78	0,028	1,55	1,4	0,056	2,8	1,9	0,095	3,8
5	0,82	0,033	1,62	1,45	0,058	2,9	2,05	0,1025	4,1
Average	0,83	0,0316	1,654	1,53	0,0612	3,06	2,05	0,1025	4,1
Span	0,12	0,007	0,25	0,3	0,012	0,6	0,3	0,015	0,6