SLAM in Navigation Systems of Autonomous Mobile Robots*

Liudmyla Hanenko[1,†], Kamila Storchak[2,†], Svitlana Shlianchak[1,†], Maksym Vorohob[3,*,†] and Mylana Pitaichuk[3,†]

[1] *Volodymyr Vynnychenko Central Ukrainian State University, 1 Shevchenka str., 25000 Kropyvnytskyi, Ukraine*
[2] *State University of Information and Communication Technologies, 7 Solomyanska str., 03110 Kyiv, Ukraine*
[3] *Borys Grinchenko Kyiv Metropolitan University, 18/2 Bulvarno-Kudriavska str., 04053 Kyiv, Ukraine*

Abstract

SLAM (Simultaneous Localisation and Mapping) is a fundamental technology in robotics that allows autonomous systems to simultaneously create a map of an unknown environment and determine its location in it. This paper provides a detailed analysis of SLAM algorithms: EKF SLAM (Extended Kalman Filter), FastSLAM, Graph SLAM, SEIF SLAM (Sparse Extended Information Filter), LIDAR SLAM, VSLAM (Visual SLAM) and IMU SLAM. The advantages and disadvantages of these algorithms are considered. The EKF SLAM, FastSLAM, Graph SLAM, and SEIF SLAM algorithms are evaluated by key metrics such as mapping accuracy, localization accuracy, computational complexity, scalability, and convergence speed. Based on the evaluation of SLAM algorithms by these metrics, we compare their performance in different environments and conditions. EKF SLAM, which uses an extended Kalman filter, provides high accuracy but suffers from high computational complexity and sensitivity to linearisation errors. FastSLAM solves some of these problems by using a particle filter to estimate the robot's trajectory, which reduces the computational load while maintaining high accuracy. Graph SLAM formulates the SLAM problem as a graph optimization problem, which allows for more efficient data association and loop closure handling, although it increases memory usage. SEIF SLAM, using sparse information matrices, balances accuracy and computational efficiency, making it suitable for large environments. LIDAR SLAM provides very high accuracy and robustness in mapping, but its reliance on expensive sensors is a significant drawback. VSLAM uses cameras to collect data, making it less dependent on sophisticated sensors, but vulnerable to changes in lighting and environmental textures. IMU SLAM integrates data from inertial measurement devices, which increases robustness to fast movements but can accumulate errors over time. Based on a comparison of key metrics, the optimal use of each algorithm is suggested depending on the specific conditions.

Keywords

mobile robots, path planning, SLAM, algorithm, sensors, navigation

1. Introduction

The modern development of robotics requires mobile robots to have high autonomy and navigation accuracy in various environments. One of the key technologies that allows achieving such characteristics is SLAM (Simultaneous Localization and Mapping). Over the past decades, this technology has been rapidly improving. Interest in SLAM has grown due to a wide range of applications—from industrial and warehouse robots to autonomous vehicles and drones. Despite significant research progress, there are many challenges associated with increasing the accuracy and stability of SLAM in real conditions, in particular in dynamic and complex environments.

2. Analysis of the last research and publications

SLAM problem was first introduced in 1986 by C. Mitt and P. Cheeseman [1]. They proposed the extended Kalman filter EKF algorithm in the context of feature-based mapping with point landmarks and known data association. P. Newmann [2] proved in his paper that the EKF

* *CPITS 2025: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, February 28, 2025, Kyiv, Ukraine*

* Corresponding author.
† These authors contributed equally.
hanenkoliudmyla@gmail.com (L. Hanenko); kpstorchak@ukr.net (K. Storchak); shlanchaksveta@gmail.com (S. Shlianchak); m.vorohob@kubg.edu.ua (M. Vorohob); m.sablina@kubg.edu.ua (M. Pitaichuk)
0000-0003-2219-8196 (L. Hanenko); 0000-0001-9295-4685 (K. Storchak); 0000-0001-9893-5709 (S. Shlianchak); 0000-0001-5160-7134 (M. Vorohob); 0000-0001-9452-1867 (M. Pitaichuk)

converges for linear SLAM problems, where the motion model and the observation model are linear functions with Gaussian noise.

Over the past decades, the SLAM problem has attracted the attention of many researchers. S. Juler et al. [3] studied the impact of nonlinear models on EKF performance. M. Montemerlo et al. presented the FastSLAM algorithm [4], which differed from the traditional EKF SLAM at the time. The algorithm was based on recursive Monte Carlo sampling and particle filtering, and for the first time, a nonlinear process model was demonstrated. G. Grisetti and R. Kummerle [5] presented a graph-based SLAM method. The authors proposed an algorithm based on least-squares error minimization.

Modern research is actively using machine learning to improve the efficiency of SLAM. B. Beskos et al. [6] developed an ORB-SLAM-based imaging system using dynamic moving object detection using multi-view geometry and deep learning. S. Li et al. [7] applied a recurrent convolutional neural network (RCNN) to a mobile robot equipped with 2D LIDAR and an inertial measurement unit (IMU) to solve the problem of accuracy degradation at large turning angles in LIDAR SLEM.

There is currently significant progress in the development of SLAM algorithms, including traditional methods, visual approaches, and hybrid technologies. Modern sensor technologies, including LIDAR, stereo and RGB-D cameras, and IMUs, provide the high-quality data required for effective SLAM. At the same time, the use of machine learning opens up new opportunities to improve the accuracy and adaptability of SLAM.

3. Purpose and research objectives

The purpose of this work is to study the effectiveness of using SLAM (Simultaneous Localization and Mapping) technology in navigation systems of autonomous mobile robots, identify key advantages and disadvantages, and determine the prospects for the implementation of SLAM and its further development. Let’s outline the research objectives:

- Analyze SLAM algorithms, their advantages and disadvantages.
- Consider the use of various sensors (LIDAR, cameras, IMU) in SLAM systems.
- Perform a comparative analysis of algorithms based on key metrics.
- Determine the optimal use of each algorithm depending on specific conditions and requirements for robotic systems.
- To identify promising areas of further research in the field of SLAM.

4. Results research

The navigation system of an autonomous mobile robot contains four key parts: localization, perception, planning, and control. Localization is the process of estimating the position of the mobile robot relative to a coordinate system or map. The perception system monitors the environment around the robot and identifies obstacles. By determining the coordinates of objects in the environment, a map is created. Path planning is the stage that uses localization and perception information to determine the optimal path in subsequent movement epochs. This plan is then translated into action by the components of the control system [8].

SLAM is a mapping technology with simultaneous localization of a mobile robot based on current sensor measurements [9]. By localization, we mean confirming the location of the mobile robot and surrounding objects in the world coordinate system, and by mapping, we mean creating a map of the environment perceived by the mobile robot [10].

SLAM enhances a robot’s ability to interpret its environment and interact effectively with it [11]. This technology is used in cases where the robot has no access to a map of the environment or precise information about its location. Only sensor measurements $z_{1:t}$and control signals are known $u_{1:t}$.

SLAM technology is presented in Fig. 1.

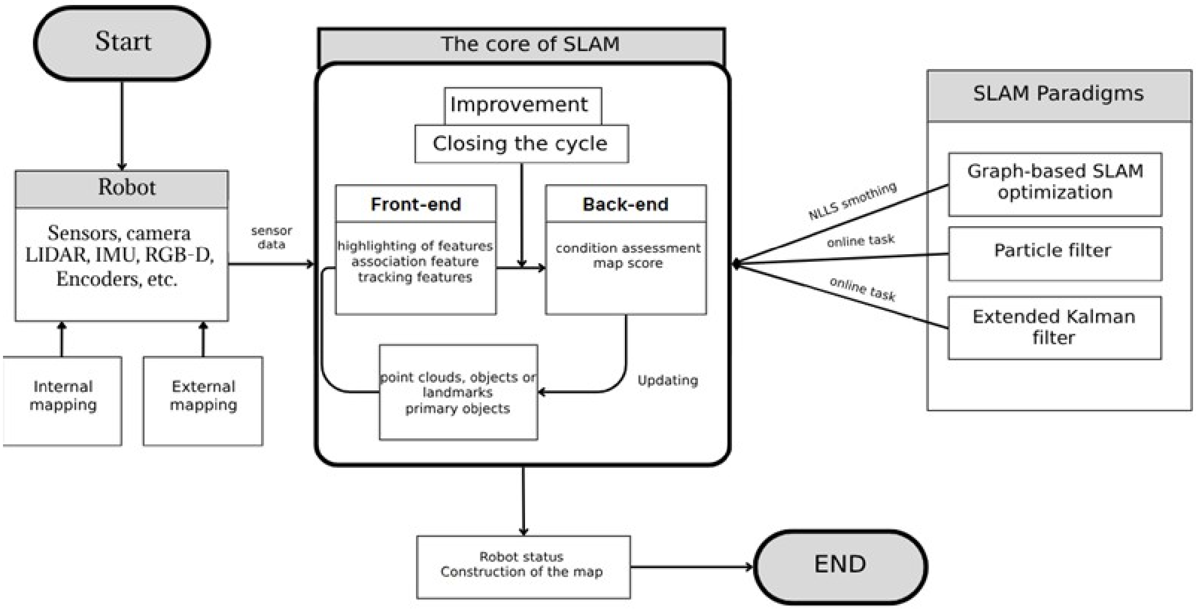

Figure 1: SLAM implementation (adapted from [12])

In real-life scenarios, SLAM deals with high uncertainty of the environment and robot positions. Therefore, the SLAM problem is usually defined using probabilistic tools [12].

SLAM has two types of problems—online SLAM problems and offline (full) SLAM problems. Online SLAM problem estimates the posterior distribution of the instantaneous value of the location on the map $p\left(x_t, m \middle| z_{1:t}, u_{1:t}\right)$, where x_t is the location at time *t*, *m* is the map, $z_{1:t}$is the measurement signals, $u_{1:t}$is the control signal. In an offline SLAM problem, it is necessary to calculate the posterior probability along the entire path $x_{1:t}$and the map $p\left(x_{1:t}, m \middle| z_{1:t}, u_{1:t}\right)$. The online SLAM problem is the result of integrating all previous positions from the offline SLAM problem [13].

4.1. SLAM algorithms

SLAM algorithms are classified depending on the types of sensors, mathematical methods, map structure, dimensionality, and computational resources. The SLAM classification scheme is implemented in Fig. 2.

SLAM algorithms by sensor type. Depending on the sensor used, the following SLAM algorithms are distinguished: laser SLAM (LiDAR-SLAM), visual SLAM (vSLAM), inertial SLAM (IMU-SLAM), and combined (Sensor Fusion SLAM).

Table 1 describes examples of sensor-based algorithms, their advantages and disadvantages. **SLAM algorithms based on mathematical methods.** There are two types of SLAM based on mathematical methods: filter-based and optimization-based. Filter [8]—based SLAMs include EKF SLAM, FASTSLAM, and SEIF SLAM.

EKF SLAM. Extended Kalman Filter-based SLAM (EKF SLAM) is a standard SLAM algorithm. It is based on a Bayesian filter, in which all variables are treated as Gaussian random variables.

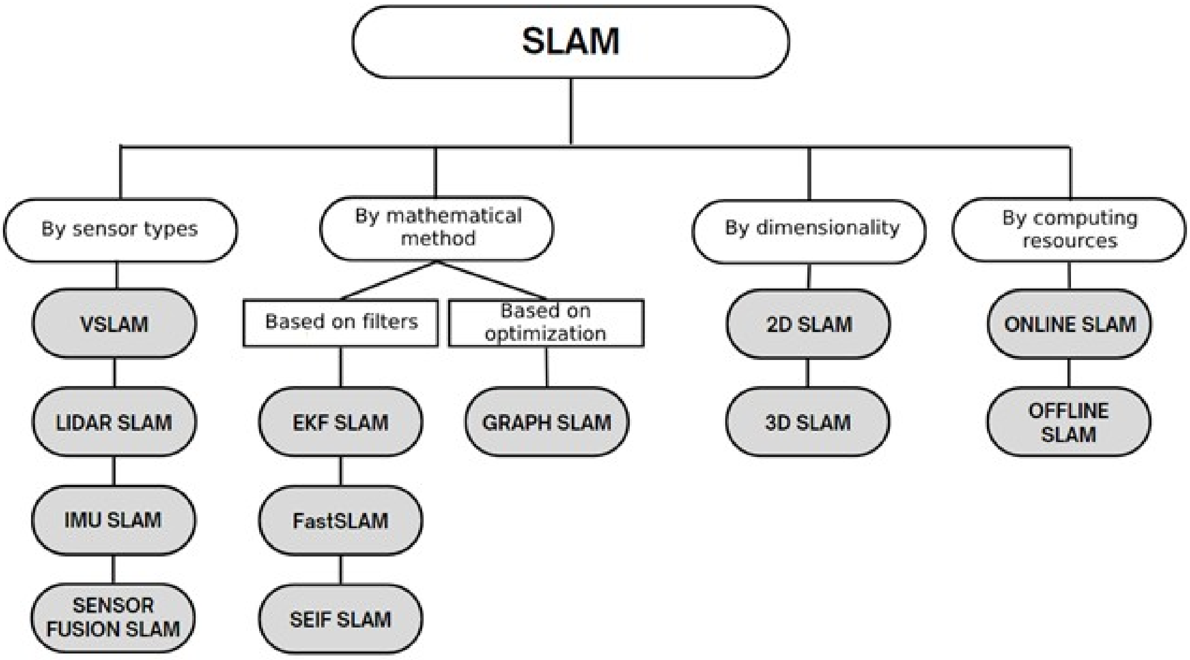

Figure 2: SLAM classification

EKF SLAM uses an extended Kalman filter to estimate the robot's position and the location of landmarks on the map. It works based on the following steps:

1. Estimation—the robot estimates its current position and orientation based on the previous state and movement patterns.
2. Update—the robot uses sensors to measure the distance to landmarks and adjusts its position and map based on these measurements.

The block diagram of the EKF algorithm SLAM is presented in Fig. 2. At time k, exteroceptive data are received and, according to formula (1), the state is predicted, and the detected landmarks are compared with those existing on the map.

$$p\left(x_{k\vee k-1}, M_{k\vee k-1} \middle| Z_{0:k-1}, U_{0:k}\right) = \int p\left(x_{k\vee k-1} \vee x_{k-1\vee k-1}, u_k\right) p\left(x_{k-1\vee k-1}, M_{k-1\vee k-1} \vee Z_{0:k-1}, U_{0:k-1}\right) d\, x_{k-1\vee k-1} \tag{1}$$

Using the recognized landmarks, the state of the mobile robot and the map are updated according to formula

$$p\left(x_{k\vee k}, M_{k\vee k} \middle| Z_{0:k}, U_{0:k}\right) = \frac{p\left(z_{i,k\vee k} \middle| x_{k\vee k-1}, M_{k\vee k-1}\right)}{p\left(x_{k\vee k-1}, M_{k\vee k-1} \middle| Z_{0:k-1}, U_{0:k}\right)} \tag{2}$$

where $x_{k\vee k-1}$ is the state vector at time k, given a known previous state $k-1$, u_k is the control vector,$M_{k\vee k-1}$ is the map estimate at time k given the previous map at time $k-1$, $U_{0:k}$ the input and control at time 0 to k, $Z_{0:k}$ is the set of observations at time 0 to k.

Table 1
Advantages and disadvantages of SLAM algorithms by sensor types

Name	Sensor	Examples of algorithms	Advantages	Disadvantages
V SLAM	Cell	ORB-SLAM, LSD-SLAM, RTAB-Map	Using high-resolution cameras allows for high-quality maps of the environment. Infrared cameras can enable vSLAM to work in low-light conditions.	Can be unreliable in sudden lighting changes, resulting in loss of accuracy, and requires significant computing power. Environments with monotone surfaces complicate the mapping and localization process.
LiDAR-SLAM	Laser rangefinder (LiDAR)	GMapping, Hector SLAM, Cartographer	Works regardless of lighting levels. Provides highly detailed 3D mapping of complex environments. Works effectively in environments with monochromatic surfaces. Data is updated at a high frequency, allowing you to quickly adapt to changes in the environment.	Sensors are expensive, large, and heavy. Processing large amounts of data requires significant computing power.
IMU-SLAM	Inertial Measurement Unit (IMU)	VINS-Mono, MSCKF	A high data update rate is provided, thanks to which the system quickly responds to changes in the robot’s movement and location. It works regardless of external conditions. The sensors are small in size and weight.	Over time, errors accumulate, and without periodic correction, the accuracy of localization and mapping decreases. They require precise calibration.
Sensor Fusion SLAM	Sensor combination	Roboception, LOAM	Allows you to compensate for the weaknesses of individual sensors. Ensures system stability in the event of a single sensor failure.	Processing data from multiple sensors requires significant computing power. It requires complex algorithms for data synchronization.

If the landmark is not found on the map, it is initialized and added to the map. This process is recursive [14].

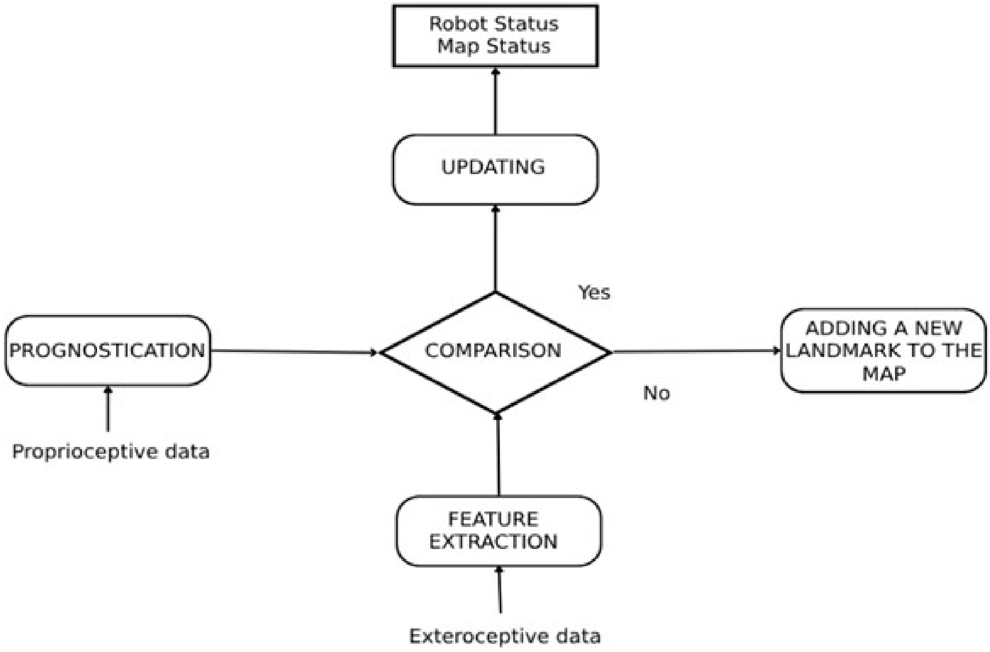

Figure 3: EKF flowchart SLAM

EKF SLAM has been successfully applied in robotics for various mapping tasks. If the landmarks are sufficiently distinguishable, the posterior estimate is computed quite well. The advantage of the full posterior estimate is its completeness, it takes into account the full uncertainty and allows the robot to evaluate the control effect according to the true value of the uncertainty.

One of the advantages of EKF SLAM is its ability to provide a good reference for loop closure. Loop closure allows the mobile robot to recognize whether it has passed the same landmark on its map. Without this, the estimation of the mobile robot's position to the landmark will be incorrect. An advantage of EKF SLAM is its ability to linearize a nonlinear model.

At each time epoch, the measurements and motion models are linearized. However, since the linearization is not performed around the true value of the state vector but around the estimated value, the linearization error will accumulate and may cause estimation divergence.

Many studies indicate that one of the main problems of EKF SLAM is its inefficiency when the map becomes larger and more complex. As the number of landmarks increases, EKF SLAM becomes slower and more computationally difficult. This is due to its computational complexity, which is quadratic. Increasing map size leads to data association problems: EKF cannot properly associate loop closures, and the nonlinearity of the environment in these maps can lead to inconsistencies and cause convergence problems [15].

A large number of algorithms have been developed to improve computational efficiency. For example, the compressed extended Kalman filter (CEKF) algorithm significantly reduces computation by focusing on local regions and then spreading the filtered information to a global map [16]. Submap algorithms have also been used to solve computational problems [17]. A new empty map is used to replace the old map when the old map reaches a predetermined size. A higher-level map is maintained to track the connection between each submap.

FastSLAM. Another class of filter-based SLAM methods is FastSLAM. FastSLAM uses a particle filter to estimate possible robot states. Each particle represents a possible robot state and the corresponding map. The algorithm has the following main steps:

1. Evaluation—each particle evaluates the new state based on the motion model.
2. Update—each particle updates the map and weight based on sensor measurements.

3. Resampling—particles with higher weights are selected for the next cycle, providing a more accurate estimate of the state.

FastSLAM—considers the robot position distribution as a set of Rao-Blackwellized particles. Using the Rao-Blackwellized filter to sample the trajectory of a mobile robot has been shown to require less memory because some particles will be removed during the update process. Since in FastSLAM, each landmark is processed separately through the EKF, it allows for more landmarks to be processed, as well as each data association based on each particle. This provides better accuracy of data association. Thus, it can reduce the loop closure problem. The computational complexity of FastSLAM is significantly reduced. Another advantage over EKF is that particle filters can handle nonlinear motion models.

FastSLAM suffers from the problem of degeneracy due to the proposal distribution process during sampling, which requires particle history. However, the FastSLAM 2.0 algorithm allows to slow down the rate of degeneration. In addition to the degeneracy problem, FastSLAM also has the disadvantage of sample depletion, and particle depletion [17].

SEIF SLAM. SEIF SLAM is based on the Extended Information Filter (EIF), which represents the state of the system as an information matrix. The main idea is that instead of working with a covariance matrix (which can be calculated quickly but has certain limitations on accuracy in complex conditions), an information matrix is used, which allows for effective uncertainty management.

The main steps of the SEIF SLAM algorithm include:

1. Condition prediction.
2. Assessment of the robot's state (position and orientation) and cartographic features, which is predicted as follows:

$$\widehat{\mu}_t = f(\mu_{t-1}, u_t) \tag{3}$$

3. Uncertainty value—information matrix Ω_t and the vector of weighted measurements ξ_t are updated at each step taking into account new measurements and control signals.
4. Feature-based assessment—an important aspect of SEIF SLAM is the preservation of cartographic features in the form of an information matrix, which allows you to preserve only important connections between features.
5. Update assessment—after receiving new measurements from sensors (e.g. LIDAR or cameras), the assessment of the state and cartographic features is updated by incrementally updating the information matrix.

SEIF SLAM is a powerful tool for SLAM that can provide high accuracy and efficiency in a variety of environments, especially where resource efficiency and measurement accuracy are important. The advantages of SEIF SLAM are effective uncertainty management due to the information matrix, the ability to operate in dynamic environments with high accuracy, and reduced computational costs compared to other SLAM methods.

Graph SLAM. Graph SLAM models the localization and mapping problem as a graph. In Graph SLAM, the positions of a mobile robot along its entire trajectory and all detected landmarks are considered as nodes of a graph. Edges on the graph connect either the robot's positions or the positions of objects that were measured there.

After the graph is constructed, graph optimization methods are applied. Methods such as Gauss-Newton or Levenberg-Marquardt are used for optimization and approximation. For graph-based SLAM, the size of its covariance matrix and the update time is constant after the graph is generated, so Graph SLAM has become popular for creating large-scale maps. [8].

The main steps of the Graph SLAM algorithm are:

1. Graph construction—the robot adds new vertices and edges to the graph based on sensor data.
2. Graph optimization—using optimization algorithms (e.g., Gauss-Newton algorithm or Levenberg-Marquardt algorithm) to minimize errors between estimated and actual distances and angles.

The advantage of Graph SLAM is the matrix structure that contains the state of the mobile robot and landmarks on the map. The large amount of information allows you to visualize the entire trajectory, which provides better accuracy in the assessment. In addition, the ability of Graph SLAM to calculate the optimal minimum cost function provides the best possible estimate of the position of the mobile robot relative to landmarks [17].

4.2. SLAM algorithms

SLAM algorithms are evaluated using various metrics that allow us to compare their effectiveness and performance in different environments and conditions. Let us outline the main metrics used to evaluate SLAM algorithms:

- Localization accuracy (how accurately the algorithm can determine the robot's position in space).
- Map construction accuracy (how accurately the algorithm can reproduce a map of the environment).
- Computational complexity.
- Reliability (algorithm's resistance to data noise, dynamic changes in the environment, data loss, etc.).
- Scalability (the ability of the algorithm to work effectively with large amounts of data and in large environments).
- Convergence (the speed of convergence of the algorithm to a stable state).

Table 2
Comparison of SLAM algorithms

Metrics	EKF SLAM	FastSLAM	Graph SLAM	SEIF SLAM
Map construction accuracy	High	High	Very high	High
Localization accuracy	High	High	Very high	High
$n \log_2 n$Computational complexity	O(n2)	O( )	O(n3)	O(n)
Reliability	Medium	High	Very high	High
Scalability	Low	High	High	High
Convergence	Fast for small cards	Ambulance	Ambulance	Ambulance

5. Discussion of research results

Evaluating SLAM algorithms by these metrics allows us to compare their effectiveness. Graph SLAM demonstrates the highest accuracy in mapping and localization due to effective optimization methods. EKF SLAM has high computational complexity due to the need to process matrices. FastSLAM and SEIF SLAM reduce computational complexity by using particle filters and sparse information matrices, respectively. SEIF SLAM demonstrates good scalability, which allows it to work effectively in large environments. Graph SLAM also scales well, but requires significant computational resources. Graph SLAM takes more time to optimize, but provides high accuracy of the final results.

The choice of a specific SLAM algorithm depends on the specific application conditions and system requirements. Using EKF SLAM can be appropriate for simple robotic systems in environments with a limited number of features and low noise. FastSLAM is effective when working with large noise and complex environments due to the use of multiple hypotheses. Graph SLAM is used in robotic systems operating in complex dynamic environments, where it is important to take into account changes in the environment and the movement of objects. If high accuracy in determining the robot trajectory and mapping is required, Graph SLAM can provide better results compared to other algorithms.

Conclusions

This paper analyzed the following simultaneous localization and mapping (SLAM) algorithms: EKF SLAM, FastSLAM, Graph SLAM, SEIF SLAM, LIDAR SLAM, VSLAM, and IMU SLAM.

Combining algorithms can significantly improve the overall performance of SLAM systems. For example, combining LIDAR SLAM and VSLAM allows you to take advantage of the advantages of both algorithms, reducing dependence on specific sensors and increasing the accuracy and reliability of the system.

Prospects for further research in SLAM include the development of new methods and approaches, as well as the improvement of existing technologies to achieve better accuracy, efficiency, and reliability. Key areas for future research are the use of machine learning methods and neural networks to improve SLAM algorithms; expanding SLAM capabilities through integration with other types of sensors; and optimizing SLAM algorithms to ensure fast processing and real-time adaptation, which will allow them to be used in a wider range of applications; improving mapping optimization algorithms to reduce accumulated error; developing optimal SLAM algorithms for use in systems with limited computing resources.

Declaration on Generative AI

While preparing this work, the authors used the AI programs Grammarly Pro to correct text grammar and Strike Plagiarism to search for possible plagiarism. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the publication's content.

References

[1] R. C. Smith, P. Cheeseman, On the representation and estimation of spatial uncertainty, Int. J. Rob. Res. 5(4) (1986) 56–68. doi:10.1177/027836498600500404

[2] P. Newman, On the structure and solution of the simultaneous localisation and map building problem, vol. 2(2) (1999) 147–158.

[3] S. Julier, J. K. Uhlmann, A counter example to the theory of simultaneous localization and map building, in: IEEE International Conference on Robotics and Automation, vol. 4, 2001, 4238–4243. doi:10.1109/ROBOT.2001.933280

[4] M. Montemerlo, et al., FastSLAM: A factored solution to the simultaneous localization and mapping problem, in: 18th National Conference on Artificial Intelligence, 2002, 593–598.

[5] G. Grisetti, et al., A tutorial on graph-based SLAM, in: IEEE Intelligent Transportation Systems Magazine, vol. 2, no. 4, 2010, 31–43. doi:10.1109/MITS.2010.939925

[6] B. Bescos et al., DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes, in: IEEE Robot Autom Lett. Institute of Electrical and Electronics Engineers Inc. 3(4) (2018) 4076–4083. doi:10.1109/LRA.2018.2860039

[7] C. Li et al., Deep sensor fusion between 2D laser scanner and IMU for mobile robot localization, Sensors 21(6) (2019) 8501–8509. doi:10.1109/jsen.2019.2910826

[8] S. Zheng, et al., Simultaneous localization and mapping (SLAM) for autonomous driving: Concept and analysis, Remote Sens. 15(4) (2023). doi:10.3390/rs15041156

[9] J. Ren, et al., Path planning algorithm and application research of an indoor substation wheeled robot navigation system, Electronics 11(12) (2022). doi:10.3390/electronics11121838
[10] W. Chen, et al., SLAM overview: From single sensor to heterogeneous fusion, Remote Sens. 14(23) (2022). doi:10.3390/rs14236033
[11] B. Al-Tawil, et al. A review of visual SLAM for robotics: Evolution, properties, and future applications, Front Robot AI 11 (2024). doi:10.3389/frobt.2024.1347985
[12] B. Alsadik, S. Karam, The simultaneous localization and mapping (SLAM)—An overview, J. Appl. Sci. Technol. Trends 2(2) (2021) 147–158. doi:10.38094/jastt204117
[13] S. Thrun, W. Burgard, D. Fox, Probabilistic robotics, 1999.
[14] C. Debeunne, D. V. Sensors, A review of visual-LiDAR fusion based simultaneous localization and mapping, Sensors 20(7) (2020). doi:10.3390/s20072068
[15] T. T. O.Takleh, et al., A brief survey on SLAM methods in autonomous vehicle, Int. J. Eng. Technol. 7(4) (2018) 38–43. doi:10.14419/ijet.v7i4.27.22477
[16] J. E. Guivant, E.M. Nebot, Optimization of the simultaneous localization and map-building algorithm for real-time implementation, in: IEEE Transactions on Robotics and Automation, vol. 17, no. 3, 2001, 242–257. doi:10.1109/70.938382
[17] M. Chli, A. J. Davison, Automatically and efficiently inferring the hierarchical structure of visual maps, in: IEEE Int. Conf. Robot. Autom., 2009, 387–394. doi:10.1109/robot.2009.5152530