https://doi.org/10.31891/2307-5732-2025-355-35
УДК 004.93

KORPAN YAROSLAV
Cherkassy State Technological University
https://orcid.org/0000-0002-1455-5977
e-mail: y.korpan@chdtu.edu.ua
NECHYPORENKO OLGA
Cherkassy State Technological University
https://orcid.org/0000-0002-3954-3796
e-mail: o.nechyporenko@chdtu.edu.ua
IVCHENKO PETRO
Cherkassy State Technological University
https://orcid.org/0000-0003-2962-977X
e-mail: p.a.ivchenko.asp22@chdtu.edu.ua

RESEARCH OF METHODS AND MEANS OF AUTOMATIC LOCATION OF A MOBILE DEVICE IN SPACE IN CONDITIONS OF PARTIAL OR FULL AUTONOMY

The rapid development of robotics, in particular mobile robots, raises a wide range of problems regarding their localization and navigation. The paper presents the results of a study of methods and means used to localize a mobile device in space. Today, mobile devices (robotic systems, drones) operate in various areas of human activity under direct control, semi-automatically or in an autonomous mode. Many localization and mapping (SLAM) methods and algorithms have been developed for their productive operation. Many of the developed methods have been successfully implemented, but they were usually developed to work in specific environments and under certain operating conditions. Therefore, a specific SLAM algorithm is selected for different conditions or, if necessary, developed/improved. This article proposes an updated and supplemented classification of SLAM algorithms. The paper classifies and summarizes some of the SLAM algorithms that have been proposed by scientists around the world and described in their publications.

A model of the localization process implementation of a mobile object in space is presented in the form of conditional blocks, which allows their compilation depending on the degree of importance and feasibility of use, and can also be used to create new SLAM algorithms. That is, the presented model allows you to determine parts of the methods of localization of an object in space, which can be replaced partially or completely, depending on the current operating conditions. Also, it is possible to switch between SLAM algorithms, which allows you to use the most optimal algorithm for specific operating conditions. For example, switching between algorithms for localization in a closed space to an algorithm for working in an open space. Switching can be both one-time and repeated.

The approaches to implementing the SLAM algorithm specified in the classification allow, if necessary, to use them in combination or separately for mapping, localization, navigation and augmented reality. Thanks to the combination of SLAM implementation approaches, mobile devices can use the most effective implementation of the localization process in their work that suits the current situation.

Keywords: mobile devices, SLAM classification

КОРПАНЬ ЯРОСЛАВ, НЕЧИПОРЕНКО ОЛЬГА, ІВЧЕНКО ПЕТРО
Черкаський державний технологічний університет

ДОСЛІДЖЕННЯ МЕТОДІВ ТА ЗАСОБІВ АВТОМАТИЧНОЇ ЛОКАЛІЗАЦІЇ МОБІЛЬНОГО ПРИСТРОЮ В ПРОСТОРІ В УМОВАХ ЧАСТКОВОЇ АБО ПОВНОЇ ЙОГО АВТОНОМНОСТІ

Стрімкий розвиток робототехніки, зокрема мобільних роботів, відриває широкий спектр завдань що до їх локалізації та навігації. В роботі наведено результати дослідження методів за засобі, які застосовуються для локалізації мобільного пристрою в просторі. На сьогоднішній день мобільні пристрої (роботизовані системи, дрони) працюють в різних сферах людської діяльності в умовах безпосереднього керування, напівавтоматично або в автономному режимі. Для їх продуктивної діяльності розроблено безліч методів та алгоритмів локалізації та картографування (SLAM). Багато з розроблених методів успішно реалізовані, але вони, як правило, розроблялися для роботи в конкретних середовищах та під певні умови роботи. Тому для різних умов підбирається або, при необхідності, розробляється/удосконалюється конкретний алгоритм SLAM. У цій статті пропонується оновлена та доповнена класифікація алгоритмів SLAM. В роботі класифіковано та узагальнено деякі з алгоритмів SLAM, які були запропоновані науковцями світу та описані в їх публікаціях. Зазначені в класифікації підходи до реалізації алгоритму SLAM дозволяють, при необхідності, застосовувати їх комплексно або окремо для картографування, локалізації, навігації та доповненої реальності. Завдяки комбінуванню підходів до реалізації SLAM мобільні пристрої в своїй роботі можуть використовувати найефективнішу реалізацію процесу локалізація, яка підходить під поточну ситуацію.

Ключові слова: мобільні пристрої, класифікація SLAM

Стаття надійшла до редакції / Received 08.05.2025
Прийнята до друку / Accepted 16.06.2025

Formulation of the problem

Simultaneous localization and mapping (SLAM) is one of the main problems of automated mobile devices (robots) that operate either in autonomous or semi-autonomous mode and consists in the ability of this object to navigate in space. As a rule, such robots are faced with two tasks, firstly, to determine what surrounds it (mapping), and secondly, to estimate its own position in the environment (localization). For this, various equipment is used, such as ultrasonic sensors, cameras or light detection and ranging devices (LIDAR), as well as various calculation algorithms.

Most localization systems are designed for static environments, and their reliability and accuracy are still questioned in dynamic systems. In such environments, SLAM algorithms are needed to deal with a certain degree of uncertainty and possible errors.

Since SLAM is used by mobile devices whose operating conditions may change during operation, for example, they may need to quickly switch from working in a closed space (room) to working in an open space, the mobile device must be able to switch between SLAM algorithms during its operation whenever the conditions change. This requires a general structuring and classification that would unite the common features and differences between the stages of SLAM implementation.

The purpose of the work is to study methods and means of automatic localization of a mobile device in space, using SLAM algorithms, in conditions of partial or complete autonomy of this device.

Analysis of recent sources

To date, scientists around the world have proposed a large number of methods for localizing objects in space. The proposed algorithms can be used for both static and dynamic objects, and each of these algorithms can be considered as a specialized solution, depending on the current situation and operating conditions. For example, in [1], different algorithms have been proposed for cases of mobile device operation in a static or dynamic environment, in open or closed space, in air or under water.

Analysis of the results of research by scientists around the world has shown that each new SLAM algorithm is either developed from scratch or with a small portion of existing algorithms. Therefore, it is difficult to classify SLAM algorithms into separate blocks that would have common features, and sometimes such a division is conditional.

In [1], a functional model of the SLAM algorithm execution path selection is presented, which is based on the analysis of commonality and variability. The presented model allows us to assess the possibility of systematically reusing parts of the SLAM stages, as well as the possibility of dynamic data exchange between SLAM nodes during calculations.

A deep study of the Visual SLAM (vSLAM) algorithm is presented [2] as a set of SLAM methods that uses the results coming from visual fixation devices. For visual fixation, cameras are used that capture images of the environment. Such devices are divided into monocular, stereo camera, omnidirectional camera (RGB-D). These devices were chosen due to the fact that they provide more information about the environment than, for example, GPS or LIDAR. The methods for implementing the described algorithms consist of three main stages: Localization, Mapping, Wayfinding.

The considered [3] localization process using SLAM algorithms is divided into three blocks: Path planning (which includes the process of Environment exploration and Obstacle detection), Loop closure detection and Active SLAM. A study of programs that used RL in SLAM was presented [3]. According to the presented material, the algorithm uses the most time for environment exploration, obstacle detection and path planning.

In the work [4] the result of the analysis of the SLAM algorithms development history is presented and the ways of their possible future further development are outlined. The types of main sensors used in SLAM implementation systems are described. The classification of the SLAM algorithm dataset is presented. The algorithms for synthesis of various methods and approaches are described.

The paper [5] presents a detailed overview of SLAM algorithms implemented for autonomous vehicle localization. It presents an in-depth overview of SLAM algorithms, highlights their advantages and disadvantages, and outlines future research directions.

Presenting main material

To obtain initial data about the environment, various types of equipment and sensors are used, from simple cameras, laser scanners to RGBD cameras (with the ability to obtain information about "depth"). The choice of the type of equipment depends on the environment in which the mobile device will operate (indoors, open space, air, underwater), as well as on what type of data needs to be obtained from the measuring/recording equipment. The data can be presented in the form of a grid map, functions, or as a point cloud.

Fig. 1 presents a model of the implementation of the localization process of a mobile object in space. The implementation of the localization process is presented in the form of conditional blocks, which allows their compilation depending on the degree of importance and feasibility of use, and can also be used to create new SLAM algorithms.

The device motion detection block includes the Motion Data function, which characterizes the possible trajectory of the mobile device and contains such components as Velocity Data (determines forward movement with circular motion), Odometry Data - sequential execution of rotational movements [1]. This block can also include the Extended Kalman Filter (EKF), which is used to determine the current situation, taking into account errors in measurements and movement based on information about the margin of stability of the equipment used.

The recording block and landmark search includes a Feature generation algorithm, which is focused on identifying characteristic features between images of the surrounding space captured by the device's cameras.

Block of work with particles includes function Particle filter, which is divided into several parts:

- Sample Motion.

- Resampling. This block includes, in particular, the KLD particle filter and the Random KLD, which randomly distributes particles in space, followed by selecting the best ones.

- Clusterer. This algorithm is used to close loops in display environments.

- Share size.

The fixation equipment block presents sensor devices. Four main types of sensors are distinguished: laser scanner, camera, RGBD-camera, D-camera. The format of the environment map depends on the selected sensor. In Fig. 1. in the map block, the maps are divided into two types: Special particle filter cards and Simple Map, which include Distributed particle grid map, Particle List, Information matrix, Pose-sensor graph, Pose graph, Graph Map, Feature based maps, Octo Map, Grid-based maps, Topological map, Image graph, Sparse matrix, Point cloud.

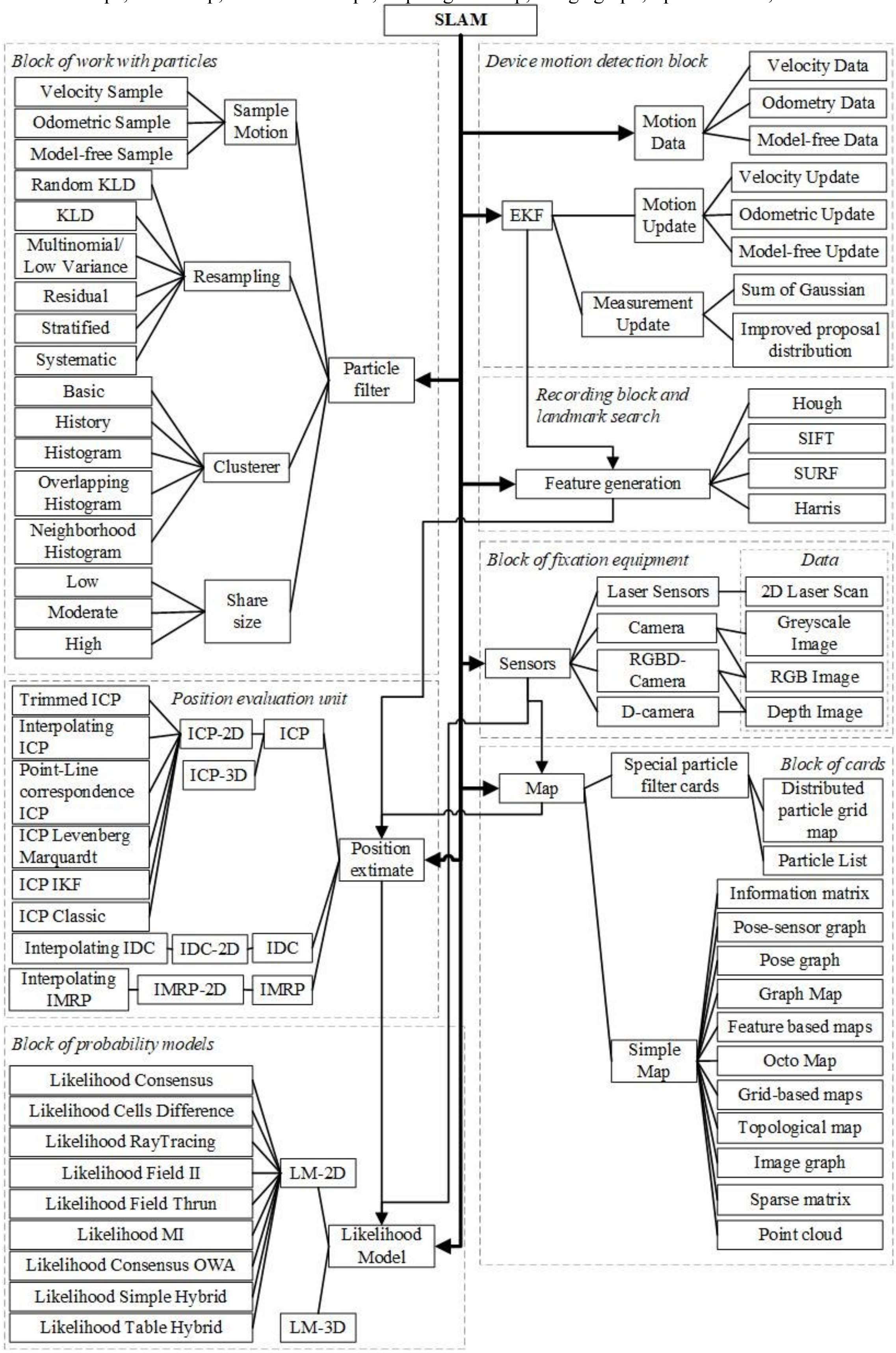

Fig. 1 Model for implementing the process of localizing a mobile object in space

The position evaluation unit presents algorithms for directly determining the position of a mobile device based on motion and sensor data. These algorithms are conditionally divided into ICP, IDC, and IMRP. The block of probability models includes probabilistic models based on the current position of the mobile object and sensor data.

The classification presented in Fig. 2 is built on the basis of common features of devices for recording data about the surrounding space, calculation algorithms and forms of map representation. The indicated algorithms can represent a map in the following form: 1 - Information matrix; 2 - Pose-sensor graph; 3 - Distributed particle grid map; 4 - Pose graph; 5 - Graph map; 6 – Feature based maps; 7 - Octo map; 8 - Grid-based maps; 9 - Topological map; 10 - Image graph; 11 - Sparse matrix.

For example, grid-based maps are two-dimensional representations where each field has a status of free, occupied, or unknown. Object-based maps contain identified objects and their spatial coordinates. Particle-based maps assume the presence of particles, which are the estimated positions of a mobile object. Graphic maps have the ability to store distances between the positions and functions of a mobile object. Graph map is the idea of dividing the environment into a grid of cells, in which the occupancy of a cell is set to free, occupied, or unknown.

Other types of maps are special cases of the presented basic types, which differ from each other mainly in memory requirements, dimensionality, and information storage.

Let us consider some algorithms presented in Fig. 2.

RatSLAM [6] is an algorithm that implements a hippocampal model that can perform real-time SLAM. It uses an attractor network to integrate odometric information with landmark detection to form a coherent representation of the environment.

RT-SLAM [7] is a standard formulation of EKF-SLAM based on visual functions such as Harris Corner. The internal architecture of RT-SLAM provides the necessary framework for flexible and efficient development of a spatial recognition system with the addition of new functionalities such as new landmark detection methods and new sensors such as echo sounder.

The modern *SeqSLAM* algorithm [8] was accelerated using the method of searching for nearest neighboring images based on the histogram of the gradient space of the descriptor, adding odometric information to the images, and using a Bayes filter to obtain a subset of sequences that are candidates for closing the loop.

Semi-Direct Visual Odometry (SVO) [9] uses feature matching. However, feature matching in this algorithm is an implicit result of direct motion estimation, rather than explicit feature extraction and matching. Feature extraction is only required when a key frame is selected to initialize new points. The advantage of SVO is increased speed due to the lack of feature detection in each frame and increased accuracy due to subpixel feature matching.

Parallel Tracking and Mapping (PTAM) [10] divides the scene tracking and mapping into parallel streams, which improves the efficiency and real-time performance of monocular systems. However, PTAM suffers from scale ambiguity problems inherent in monocular SLAM systems. Solution to this problem is presented in the Stereo Parallel Tracking and Mapping (S-PTAM) algorithm [11], which uses a stereo base to obtain depth information, which somewhat eliminates scale ambiguity and improves reliability.

Dense Tracking and Mapping (DTAM) [12] is a system for camera tracking and real-time reconstruction of the surrounding space, the work is based not on the selection of certain features of frames, but on the density of each pixel. The created model consists of depth maps built from groups of frames using dense and accurate subpixel stereo reconstruction of multiple angles.

The *Large-Scale Direct Monocular SLAM (LSD-SLAM)* algorithm [13] works with image intensity for both sensor position tracking and mapping. The camera position is tracked using direct image alignment, and the geometry is estimated in the form of depth maps obtained by filtering multiple pixel-per-pixel stereo comparisons. The position graph is constructed from keyframes, allowing for the reconstruction of large-scale maps with corrected scale drift, including scene locking.

In the initial stages of *DH-PTAM* [10], a keyframe is established based on the camera position in the initial camera state. A preliminary map is created by identifying and triangulating the distinctive points in the initial stereo pair of images. For subsequent frames, the tracking stream computes the pose of each stereo frame by minimizing the discrepancies between the predicted map points and their matches. The system selects a subset of keyframes that are used in the other stream.

The *S-PTAM* algorithm [11], using data from stereo cameras, allows visual landmarks to be matched on a pair of consecutive images, accurately reconstructing their real depth. The loop is closed by comparing images based on appearance and correcting the process by optimizing the graphical representation of the location on the map.

The *ORB-SLAM2* algorithm [14] consists of such key blocks as: Feature detection and matching, Tracking, Local Mapping, Loop Closing, Map Representation, Relocalization, Map Management and Optimization. The difference between ORB-SLAM2 and other algorithms and its strengths are: strong tracking; real-time performance; support for monocular, stereo, and RGB-D setups; map optimization and loop closure detection; scalability and map management; open-source and in active development.

The *g-mapping* algorithm [14] [15] [16] is one of the most widely used 2D-SLAM algorithms for indoor navigation. The algorithm provides a process for estimating the position of a mobile device and creating a map by filtering RBpf particles, which separates the positioning and mapping tasks.

DP-SLAM [15] is a particle filter algorithm that generates a mesh. The essence of the algorithm is to reduce the use of computational resources by avoiding the sequential copying of maps for each particle that was generated during the re-sampling stage. That is, the algorithm generates a single map, thus preserving the data structure and allowing you to always know the changes made by it and previous particles.

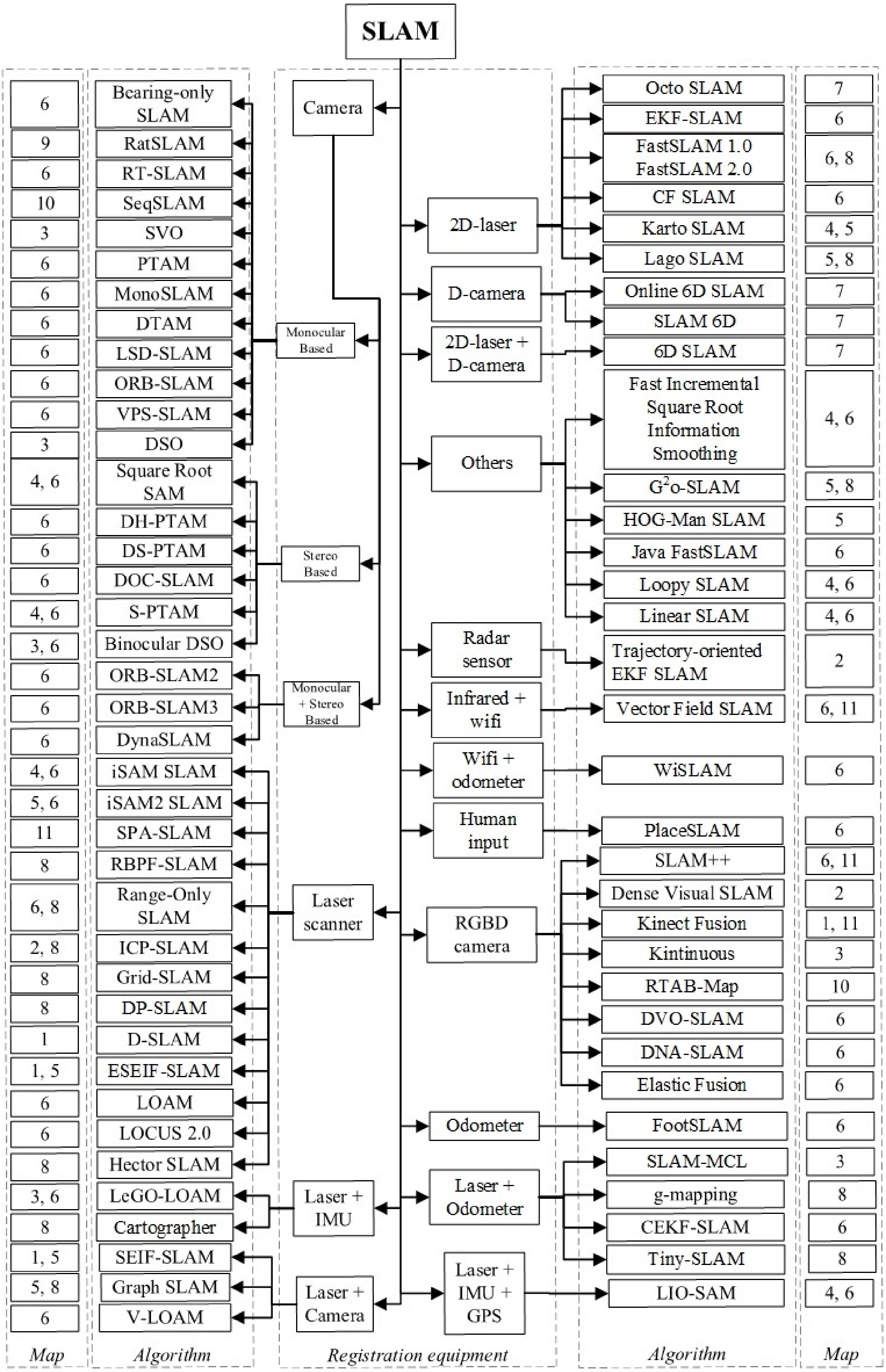

Fig. 2. Classification of SLAM algorithms according to the means of registration and the form of presentation of results

Exactly Sparse Extended Information Filte (ESEIF-SLAM) [17] takes advantage of the sparse canonical parameterization while maintaining conservative (relative to the standard Gaussian distribution) estimates of the robot pose and map. ESEIF has estimation results that are almost identical to EKF, while maintaining computational time and memory requirements. ESEIF demonstrates improved scalability while maintaining conservative estimates both globally and locally.

Hector-SLAM [14] [16] is an algorithm that uses a mapping method to map an unknown environment and select a position to determine the location of a mobile device in that environment. The algorithm starts with some initial assumption about the object's position, then refines the object's position by comparing laser scanning data with an existing map, and gradually updates the occupancy grid to correct the calculated object's position.

LeGO-LOAM [18] used when a terrain map is available in the segmentation and optimization stages. First, point cloud segmentation (for noise filtering) and feature extraction are applied to obtain characteristic planes and characteristic features. Then, the Levenberg-Marquardt optimization method uses the planar and characteristic features to solve various six-degree-of-freedom transformation problems.

Cartographer [19] is a grid-based mapping algorithm developed by Google. Unlike the original design, the algorithm has been improved somewhat, for example, it has been optimized using an adaptive multi-stage distance planner, which has accelerated the calculation time by 20% while maintaining positioning and map generation accuracy.

Combined Filter SLAM (CF SLAM) [20] is a combination of extended Kalman filters and extended information filters combined with a local mapping approach and the use of sparse Cholesky decomposition with minimal pre-ordering.

Karto SLAM [14] [16] is one of the algorithms that works on the basis of graphs for localization of a mobile object and performing mapping. Karto SLAM uses the current location of the object and a complete map of the terrain, and also adds/updates data by position. The main disadvantage of Karto SLAM is that as the number of landmarks increases, the computational complexity increases proportionally.

Lago SLAM [16] is based on Graph SLAM and is diagram-based. The essence of the algorithm is that the optimization process to find the correct graph configuration does not require an initial evaluation. However, this leads to an increase in the computational complexity of the algorithm.

Fast Incremental Square Root Information Smoothing [21] is based on incremental smoothing. The algorithm is suitable for real-time systems in large-scale environments. The efficiency is achieved by updating the square root information matrix, which is factored into a version of the regular sparse information smoothing matrix.

Kinect Fusion [22] performs localization and mapping using an inverse depth parameterization model. The method uses the depth function from Kinect to improve convergence. An extended Kalman filter (EKF) is used to estimate the camera trajectory and object positions.

The *Kintinuous* algorithm [23] relies heavily on depth data. The environment data is stored in the GPU as a truncated signed distance function (TSDF), essentially a three-dimensional array of distances and weights. The camera pose in the TSDF, i.e. the position and direction the camera is pointing, is given by a rotation and translation matrix. Unlike the Kinect Fusion algorithm, the TSDF re-centers on the camera when it moves too far from the origin, allowing for an increase in the model volume. When the TSDF is moved to a part of the point cloud that is no longer included, a mesh algorithm is applied to obtain a triangulated model.

Real Time Appearance Based Mapping (RTAB-Map) [23] is a graph-based algorithm optimized for large spaces and allows for real-time loop closure. RTAB-Map successfully copes with 3-Dimensional SLAM problems. RTAB-Map uses RGB data more efficiently than Kintinuous.

The *DVO-SLAM* algorithm [24] minimizes the photometric error between consecutive RGB-D frames by weighting the photometric errors according to a t-distribution. There are several extensions and optimizations of this algorithm, for example, for moving shutter cameras, planar scenes, and many others.

Dense Noise Aware SLAM (DNA-SLAM) [24] involves estimating the reliability of each individual pixel in a ToF RGB-D image to weight its impact on the estimation of camera motion. That is, dense noise-aware localization and mapping is performed, which takes into account the noise characteristics of ToF RGB-D cameras with a complex weighting scheme.

FootSLAM [25] is a method for converting position drift odometry into pose estimates with long-term error stability. In the FootSLAM algorithm, the two-dimensional navigation region is represented by a grid of hexagons, and then the probability of transitioning from one hexagon to the next is determined.

The *CEKF-SLAM algorithm* [15], based on the Kalman filter, creates maps using features of the surrounding environment; it also optimizes the EKF-SLAM algorithm using a compressed filter that delays the update of the covariance associated with the set of non-local labels.

Tiny-SLAM [15] is based on a single-grid high-resolution particle filter. In laser scanning, more than one landmark is updated per line incident on the surface, and a function is implemented that creates holes in the map to improve the likelihood function.

Graph SLAM [16] addresses the weaknesses of particle filter (FastSLAM) and EKF SLAM. The algorithm involves creating a graph whose nodes represent the position of the mobile device or control points. After creating this graph, a node configuration is found that will have the smallest error with respect to the measured data. This approach allows the map to be modified after individual parts have been created. However, this can increase the computational requirements.

Conclusions

The paper presents the results of the analysis of methods and means of localization of a mobile device based on SLAM algorithms. The results of the study were systematized according to the common fundamental characteristics of the elements of the system blocks that implement the SLAM algorithm.

A model of the localization process implementation of a mobile object in space is presented in the form of conditional blocks, which allows their compilation depending on the degree of importance and feasibility of use, and can also be used to create new SLAM algorithms. That is, the presented model allows you to determine parts of the methods of localization of an object in space, which can be replaced partially or completely, depending on the current operating conditions. Also, it is possible to switch between SLAM algorithms, which allows you to use the most optimal algorithm for specific operating conditions. For example, switching between algorithms for localization in a closed space to an algorithm for working in an open space. Switching can be both one-time and repeated.

A classification of SLAM algorithms is presented, built on the basis of common features of devices for recording data about the surrounding space, calculation algorithms and map representation forms. The algorithms specified in the classification for implementing the SLAM algorithm allow, if necessary, to be used separately for mapping, localization, navigation and augmented reality.

References

1. Werner, C., Werner, S., Schöne, R., Götz, S. and Aßmann, U. Self-adaptive Synchronous Localization and Mapping using Runtime Feature Models. DOI: 10.5220/0006945504090418 In Proceedings of the 7th International Conference on Data Science, Technology and Applications (DATA 2018), pages 409-418.

2. Charalambos Theodorou, Vladan Velisavljevic, Vladimir Dyo, Fredi Nonyelu, Visual SLAM algorithms and their application for AR, mapping, localization and wayfinding, Array, Volume 15, 2022, 100222, ISSN 2590-0056, https://doi.org/10.1016/j.array.2022.100222.

3. Dehghani Tezerjani, Mohammad & Khoshnazar, Mohammad & Tangestanizade, Mohammadhamed & Yang, Qing. (2024). A Survey on Reinforcement Learning Applications in SLAM. https://doi.org/10.48550/arXiv.2408.14518.

4. Chen, W., Wang, X., Gao, S., Shang, G., Zhou, C., Li, Z., Xu, C., & Hu, K. (2023). Overview of Multi-Robot Collaborative SLAM from the Perspective of Data Fusion. Machines, 11(6), 653. https://doi.org/10.3390/machines11060653/

5. Bala, J.A.; Adeshina, S.A.; Aibinu, A.M. Advances in Visual Simultaneous Localisation and Mapping Techniques for Autonomous Vehicles: A Review. Sensors 2022, 22, 8943. https://doi.org/10.3390/s22228943

6. Milford, Michael & Wyeth, Gordon & Prasser, D.. (2004). RatSLAM: A hippocampal model for simultaneous localization and mapping. IEEE Int. Conf. Rob. Autom. 1. 403 - 408 Vol.1. 10.1109/ROBOT.2004.1307183.

7. M. Meireles, R. Lourenço, A. Dias, J. M. Almeida, H. Silva and A. Martins, "Real time visual SLAM for underwater robotic inspection," *2014 Oceans - St. John's*, St. John's, NL, Canada, 2014, pp. 1-5, doi: 10.1109/OCEANS.2014.7003097.

8 Tsintotas, Konstantinos & Bampis, Loukas & Rallis, Stelios & Gasteratos, Antonios. (2019). SeqSLAM with Bag of Visual Words for Appearance Based Loop Closure Detection: Proceedings of the 27th International Conference on Robotics in Alpe-Adria Danube Region (RAAD 2018). 10.1007/978-3-030-00232-9_61.

9. Forster, Christian & Pizzoli, Matia & Scaramuzza, Davide. (2014). SVO: Fast Semi-Direct Monocular Visual Odometry. Proceedings - IEEE International Conference on Robotics and Automation. 10.1109/ICRA.2014.6906584.

10. Soliman, Abanob & Bonardi, Fabien & Sidibé, Désiré & Bouchafa, Samia. (2023). DH-PTAM: A Deep Hybrid Stereo Events-Frames Parallel Tracking And Mapping System. 10.48550/arXiv.2306.01891.

11. Taihú Pire, Thomas Fischer, Gastón Castro, Pablo De Cristóforis, Javier Civera, Julio Jacobo Berlles, S-PTAM: Stereo Parallel Tracking and Mapping, Robotics and Autonomous Systems, Volume 93, 2017, Pages 27-42, ISSN 0921-8890, https://doi.org/10.1016/j.robot.2017.03.019.

12. Newcombe, Richard A. et al. “DTAM: Dense tracking and mapping in real-time.” 2011 International Conference on Computer Vision (2011): 2320-2327.

13. Engel, Jakob & Schoeps, Thomas & Cremers, Daniel. (2014). LSD-SLAM: large-scale direct monocular SLAM. Eur. Conf. Comput. Vis.. 8690. 1-16. 10.1007/978-3-319-10605-2_54.

14. Manar, Aicha & Haouala, Ines & Raisov, Amanzhol. (2023). Low cost mobile navigation using 2D-SLAM in complex environments. https://doi.org/10.13140/RG.2.2.31842.50887.

15. R. Lemus, S. Díaz, C. Gutiérrez, D. Rodríguez and F. Escoba (2014) SLAM-R Algorithm of Simultaneous Localization and Mapping Using RFID for Obstacle Location and Recognition // Journal of Applied Research and Technology P. 551-559.

16. František Duchoň, Jakub Hažík, Jozef Rodina, Michal Tölgyessy, Martin Dekan, Adam Sojka (2019) Verification of SLAM Methods Implemented in ROS. Journal of Multidisciplinary Engineering Science and Technology (JMEST) ISSN: 2458-9403 Vol. 6 Issue 9, September – 2019 p. 10579-10584

17. Wang, Zhan & Huang, Shoudong & Dissanayake, Gamini. (2007). Tradeoffs in SLAM with Sparse Information Filters. Springer Tracts in Advanced Robotics. 42. 10.1007/978-3-540-75404-6_32.

18. T. Shan and B. Englot, "LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, 2018, pp. 4758-4765, doi: 10.1109/IROS.2018.8594299.

19. Dwijotomo, A.; Abdul Rahman, M.A.; Mohammed Ariff, M.H.; Zamzuri, H.; Wan Azree, W.M.H. Cartographer SLAM Method for Optimization with an Adaptive Multi-Distance Scan Scheduler. *Appl. Sci.* **2020**, *10*, 347. https://doi.org/10.3390/app10010347

20. Cadena, Cesar & Neira, José. (2009). SLAM in O(log n) with the Combined Kalman - Information Filter. (2009) IEEE/RSJ International Conference on Intelligent Robots and Systems, October 11-15, 2009, St. Louis, MO, USA2069-2076. DOI:10.1109/IROS.2009.5354521.

21. Kaess, M., Ranganathan, A., & Dellaert, F. (2007). Fast Incremental Square Root Information Smoothing. Conference Paper, Proceedings of 20th International Joint Conference on Artificial Intelligence (IJCAI '07), pp. 2129 - 2134, *2007*

22 Li, H. & Lin, Y. & Liu, H. & Yang, D.. (2013). Method of SLAM with Kinect. Zhongnan Daxue Xuebao (Ziran Kexue Ban)/Journal of Central South University (Science and Technology). 44. 28-32.

23 Altuntas, Nihal & Uslu, Erkan & Çakmak, Furkan & Amasyali, Mehmet & Yavuz, Sırma. (2017). Comparison of 3-dimensional SLAM systems: RTAB-Map vs. Kintinuous. 99-103. 10.1109/UBMK.2017.8093567.

24. Wasenmüller, Oliver & Ansari, Mohammad & Stricker, Didier. (2016). DNA-SLAM: Dense Noise Aware SLAM for ToF RGB-D Cameras. 10.1007/978-3-319-54407-6_42.

25. Wahlström, Johan & Markham, Andrew & Trigoni, Niki. (2019). FootSLAM meets Adaptive Thresholding. 10.48550/arXiv.1911.00420.