Proceedings of CONF-MLA 2025 Symposium: Applied Artificial Intelligence Research
DOI: 10.54254/2755-2721/2025.BJ26928

Introduction to Low-cost and Low-consumption Optimization Schemes for Simultaneous Localization and Mapping

Buning Jiang

School of Aberdeen Institute of Data Science and Artificial Intelligence, South China Normal University, Foshan, China
xpenetc@gmail.com

Abstract. As an important technology required for intelligent devices such as autonomous driving and autonomous robots that can actively obtain environmental information, synchronous positioning and map construction technology has developed in response to practical demands and technological progress, and has given rise to a variety of algorithms with different efficiencies and consumption levels. However, the existing research related to SLAM is more focused on improving the performance of simultaneous localization and mapping algorithms in complex environments, while there are relatively few specialized optimizations for devices with lower computing or sensing capabilities. This paper mainly studies the existing solutions for reducing the requirements of synchronous positioning and map construction technology for the computing power or perception ability of devices, sorts out the relevant academic papers since 2009, and analyzes these papers through the review method. And from this, several optimization schemes for reducing the consumption of synchronous positioning and map construction algorithms have been sorted out, such as adding additional sensors like inertial measurement units, reducing the consumption of the search process through more efficient heuristic algorithms, lowering the data processing volume through dimensionality reduction, and reducing the recognition difficulty through strong features.

Keywords: SLAM, cost, robot technology, optimization

1. Introduction

With the development of technologies such as autonomous driving and autonomous robots, the Simultaneous Localization and Mapping (SLAM) algorithm, which helps devices determine their own position and posture while judging and modeling the surrounding environment, has gradually attracted widespread attention. This algorithm acquires information about the surrounding environment by the sensors like camera or LightLaser Detection and Ranging (LiDAR) at each time point. It combines this information to obtain a model of the surrounding environment and the status of itself, like its attitude and position in the environment model. The entire process takes place on the computer carried by the device and does not require the guidance of external devices, which makes the devices using SLAM algorithm have lower requirements for infrastructure and can maintain excellent performance when running in unknown environments. Therefore, it is often used

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2025.BJ26928

in scientific research, mapping and autonomous driving. In recent years, the newly proposed optimized SLAM algorithm has higher accuracy and robustness, and can still provide an accurate model in case of insufficient surrounding information, such as a lack of lighting and key points. However, along with this comes increasingly higher demands for computing power and memory, which significantly increase the cost of the equipment and also pose greater challenges to its processing capabilities.

Reducing the demand for computing power and memory for the SLAM algorithm from the aspects of sensors and algorithms can enable this algorithm to run stably on computers with simple structures, thereby lowering the production and maintenance costs of automated unmanned operation platforms such as robots and drones that adopt this algorithm. This, in turn, helps to reduce the cost of industrial production that uses unmanned operation platforms, scientific exploration and social life.

At present, some studies have focused on reducing the consumption of SLAM, but the number of related studies is limited. This paper will be based on the existing SLAM optimizations and compare the optimization principles, applicable scenarios and optimization effect of different SLAM optimization methods through literature analysis and review. The aim of this research is to help engineers, scientists and other related practitioners who need to use SLAM to provide positioning for automatic robots and also to formulate appropriate optimization plans for navigation to reduce the cost of robots.

2. Basic SLAM algorithm

A basic SLAM algorithm framework includes the ability of Initialization, Tracking and Mapping. Initialization is the function of defining a global coordinate system to assist the camera in pose estimation and 3D reconstruction in an unknown environment. This also makes it possible to format an initial map. Tracking is responsible for estimating the possible position and posture of the camera when the image captured by the camera appears on the map. To achieve this, Tracking needs to use algorithms like Perspection-N-Point (PNP) to calculate the rotation matrix and the translation vector of the camera based on the known three-dimensional space points and their projection points on the plane, as well as the internal parameter matrix of the camera, and update the camera state. Mapping is used to expand the map in the area that has not been mapped.

To maintain stability and good performance, some improved methods are also often added to the SLAM algorithm. These methods are mainly used to process the data generated by the above steps to make them perform better and correct the resulting errors. Loop detection is one of those methods. After identifying that the device has passed by the same location again, the algorithm will adjust the previously generated data to increase the accuracy and consistency of the global map. In addition, these improved methods will also include some backup positioning means. For instance, when the algorithm fails to track the camera status, these backup means will attempt to re-obtain the camera status to prevent the algorithm from crashing [1].

3. Optimization plan

The existing SLAM computing power optimization schemes mainly focus on optimizing one or more modules of the above-mentioned SLAM algorithms, and mainly concentrate on optimizing the Tracking part. For instance, the improved SLAM framework, Fast-LIO, incorporates an Inertial Measurement Unit (IMU). It integrates the IMU's measurement results with the feature points extracted by the LiDAR through a tightly coupled iterative Kalman filter. Moreover, it employs a

DOI: 10.54254/2755-2721/2025.BJ26928

new Kalman gain calculation formular to reduce the additional computational load resulting from the increase of feature points. These changes successfully reduce the overall computing power demand and computing time [2]. Based on the Fast-LIO framework, Pang et al. incorporated the Iterative Closest Point (ICP) module which integrates the optimization methods based on the fast nearest neighbor search of KD-Tree and Gauss-Newton iterative method into the algorithm [3]. This innovation helps their SLAM algorithm achieve fast, low-consumption and accurate point cloud alignment, which helps them achieve a better result of a lower consumption of CPU and memory than the original Fast-LIO framework in their test. Compared with using ICP for point cloud registration, the DeepVCP proposed by Lu et al. adopts neural networks to avoid the Random Sample Consensus (RANSAC) that requires a large computational overhead [4]. This method uses an end-to-end network to generate virtual points, which eliminates the need for iteration and random sampling. As for the consumption of neural network, DeepVCP adopts the Singular Value Decomposition (SVD) operator based on TensorFlow for GPU acceleration. This optimization method enables neural networks to perform parallel computing using GPUs. However, DeepVCP still has the drawback of relying on GPU devices and having a relatively low generalization ability, which makes it have certain requirements for the usage environment.

Unlike the above approach, Pang et al. avoid the huge consumption of ICP in the 3D environment by projecting the point cloud in the 3D space onto the 2D plane [5]. This method not only makes it possible to reduce the computational load brought by an additional dimension but also enables the framework to utilize the rich performance improvement methods for 2D ICP. To ensure that this process of dimensionality reduction projection does not affect the recognition accuracy, this optimization method distinguishes points on the ground from those in space. By all of these efforts, this algorithm has the capability to distinguish whether the mapped points can form feature edges and feature planes. The equipment adopting this method achieved excellent results in the loop closure test in large-scale scenarios, demonstrating its reliability in large outdoor environments. VinySLAM, on the other hand, focuses on large indoor scenes [6]. It is based on a highly simplified SLAM framework called tinySLAM, which uses single-hypothesis tracking that only maintains one optimal device position and pose estimation to significantly reduce code volume and save computing resources [7]. VinySLAM uses the Transferable Belief Model (TBM) to evaluate the possibility of the state, which greatly enhances its own robustness. This algorithm performs well in low-resource scenarios and its mapping capability is significantly superior to that of its base, tinySLAM.

The above-mentioned optimization method is mainly aimed at LiDAR SLAM. Its advantage lies in the simple principle of the laser rangefinder and the more accurate data acquisition, thus having higher modeling accuracy. However, at present, the cost of laser navigation distance sensors is still relatively high. Therefore, on some more affordable devices, SLAM algorithms customized for other types of sensors, which are different from LiDAR SLAM, are often used, and these algorithms also have their own optimization methods. VSLAM is designed for devices that use cameras as the main sensor. Ordinary cameras have the advantage of being inexpensive, but they have the disadvantages of being difficult to accurately determine distances and having high difficulty in feature recognition. To achieve synchronous positioning and map drawing through cameras, it is first necessary to overcome the difficulties brought by identifying indistinct features through the relevant technologies of computer vision, and more importantly, to deal with the lens distortion caused by movement. Zhu et al. 's CamVox solves the problems of distance measurement and motion distortion by attaching a low-cost, non-repetitive scanning Livox lidar to the camera sensing system [8]. The correction of camera images by lidar and IMU enables CamVox to achieve higher accuracy and lower

DOI: 10.54254/2755-2721/2025.BJ26928

configuration requirements in outdoor large scenes than traditional visual SLAM, and its hardware cost is significantly lower than that of solutions that fully adopt this type of lidar. Peng et al. Adopted the embedded GPU design, utilizing the parallel computing capability of GPU, more lightweight algorithms and more reasonable power consumption allocation to reduce the pressure on hardware, thereby reducing the hardware requirements of the VSLAM algorithm and thus lowering the cost [9]. In contrast to the above-mentioned methods, Lee specifically optimizes the equipment by making it more adaptable to the characteristics of the working scenarios [10]. This optimization plan is entirely based on the features of low-cost consumer-grade robots, such as vacuum cleaners, and the characteristics of their working scenarios, while giving up the effect of making the equipment work in other environments. This strategy is based on a definite indoor environment. While compressing the resolution of the images captured by the camera to reduce the processing load of the CPU, it also utilizes features such as the corner lines and ceilings inside the room to enable feature point matching to be achieved with less computational effort.

In addition, there are also solutions that provide information input for SLAM algorithms through less common sensors, making it possible to apply cheaper sensors in autonomous driving devices. For example, the scheme of Yap et al. adopts sonar as the sensor. Regarding the inherent sparsity and noise problems when sonar is used as the sensor, this scheme has greatly improved them through the randomized Hough transform (RHT) algorithm [11]. According to the paper, this device that uses sonar as the sensor still has excellent accuracy even in a larger environment. This research achievement demonstrates that non-precise positioning sensors can also be used as the main sensors for SLAM, bringing more possibilities to the selection of sensors for autonomous driving devices.

4. Conclusion

This paper mainly analyzes the optimization methods and ideas of several existing SLAM algorithm optimization schemes, including the optimization methods for common but relatively expensive laser sensors and those for relatively inexpensive camera or sonar sensors. And methods such as adding additional sensors like inertial measurement units, reducing the consumption of the search process through more efficient heuristic algorithms, lowering the data processing volume through dimensionality reduction, and reducing the recognition difficulty through strong features have been summarized. However, this paper does not include the analysis and comparison of the specific effects of each optimization method on the data, and thus could not determine the specific optimization effects and advantages and disadvantages of each optimization method. Therefore, there are deficiencies in helping engineers or other relevant practitioners make specific scheme choices. Furthermore, this paper does not cover enough related research, which makes it lack comprehensiveness. In future research, researchers can attempt to test the optimization effects of mainstream SLAM optimization methods, including equipment cost, positioning and mapping accuracy, and computing resource consumption, under the same and stable environment, to determine the advantages and disadvantages of different optimization schemes in the specified environment. The above features can also be tested in various environments to evaluate the adaptability of these optimization schemes in different scenarios.

References

[1] Taketomi, T., Uchiyama, H., & Ikeda, S. (2017). Visual SLAM algorithms: A survey from 2010 to 2016. IPSJ transactions on computer vision and applications, 9(1), 16.

[2] Xu, W., & Zhang, F. (2021). Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter. IEEE Robotics and Automation Letters, 6(2), 3317-3324.

DOI: 10.54254/2755-2721/2025.BJ26928

[3] Pang, C., Zhou, L., & Huang, X. (2024). A low-cost 3D SLAM system integration of autonomous exploration based on fast-ICP enhanced LiDAR-inertial odometry. Remote Sensing, 16(11), 1979.
[4] Lu, W., Wan, G., Zhou, Y., Fu, X., Yuan, P., & Song, S. (2019). Deepvcp: An end-to-end deep neural network for point cloud registration. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 12-21).
[5] Pang, C., Tan, Y., Li, S., Li, Y., Ji, B., & Song, R. (2019, August). Low-cost and high-accuracy LiDAR SLAM for large outdoor scenarios. In 2019 IEEE International Conference on Real-time Computing and Robotics (RCAR) (pp. 868-873). IEEE.
[6] Huletski, A., Kartashov, D., & Krinkin, K. (2017, September). Vinyslam: an indoor slam method for low-cost platforms based on the transferable belief model. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 6770-6776). IEEE.
[7] Steux, B., & El Hamzaoui, O. (2010, December). tinySLAM: A SLAM algorithm in less than 200 lines C-language program. In 2010 11th International Conference on Control Automation Robotics & Vision (pp. 1975-1979). IEEE.
[8] Zhu, Y., Zheng, C., Yuan, C., Huang, X., & Hong, X. (2021, May). Camvox: A low-cost and accurate lidar-assisted visual slam system. In 2021 IEEE International Conference on Robotics and Automation (ICRA) (pp. 5049-5055). IEEE.
[9] Peng, T., Zhang, D., Hettiarachchi, D. L. N., & Loomis, J. (2020). An evaluation of embedded GPU systems for visual SLAM algorithms. Electronic Imaging, 32, 1-6.
[10] Lee, S., & Lee, S. (2013). Embedded visual SLAM: Applications for low-cost consumer robots. IEEE Robotics & Automation Magazine, 20(4), 83-95.
[11] Yap, T. N., & Shelton, C. R. (2009, May). SLAM in large indoor environments with low-cost, noisy, and sparse sonars. In 2009 IEEE International Conference on Robotics and Automation (pp. 1395-1401). IEEE.