The Development and Analysis of 3D Feature Reconstruction Technology for Service Robot SLAM System in Restaurant Environment

Zibo Zheng

University of Nottingham Ningbo China, Ningbo City, Zhejiang Province, 315000, China
ssyzz32@nottingham.edu.cn

Abstract. Indoor mobile robots are now widely used in restaurants for delivery services to improve delivery efficiency and reduce labor costs. Simultaneous visual localization and mapping (SLAM) and path planning are the basis for restaurant service robots to navigate and deliver food. Therefore, it is useful to summarize the framework and specific methods of SLAM for the development of restaurant service robots and even the service industry. In this paper, SLAM is divided into vision SLAM and LIDAR SLAM to summarize the framework and introduce and compare the specific methods. Firstly, the research background and significance of mobile robots and SLAM technology in restaurant environments are introduced. Secondly, visual SLAM & LiDAR information perception and 3D reconstruction technologies are introduced separately. Subsequently, two commonly used backend optimization methods are summarized, and the classification and construction methods of maps are summarized. Finally, the direction and opportunities for future SLAM research are discussed, and a summary of the entire article is provided. This paper provides methodological guidance for mobile robots working in a restaurant environment.

Keywords: simultaneous localization and mapping; service robot; restaurants environment; feature reconstruction; back-end optimization

1 Introduction

Recently, due to the growth of robotics, service robots have been put into many service fields such as restaurants, hotels, and navigation guides in huge buildings. Service robots are intelligent systems composed of machinery and electronics that serve human daily life [1]. The function of a service robot is to use sensors to recognize its surroundings and perform the tasks given in the corresponding scenario. Specifically, in a restaurant environment, a service robot should be able to deliver food to the consumer's table in the shortest possible time. Therefore, a satisfactory service robot should have the basic function of intelligent navigation. The key technologies of intelligent navigation mainly include autonomous positioning of robots, map construction, path planning, etc. Although a certain technological foundation is built on autonomous navigation

Y. Wang (ed.), *Proceedings of the 2024 2nd International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2024)*, Advances in Computer Science Research 115,
https://doi.org/10.2991/978-94-6463-540-9_67

technology for robots, the intelligent autonomous task execution of mobile robots in real environments still faces many challenges [2].

For the navigation of service robots, localization and map building are crucial. In the city, most robots can be localized and map building by GPS. However, due to the limitations of indoor environments where GPS does not work well, service robots need to achieve localization and map building through simultaneous localization and mapping (SLAM) technology. Localization and map building is a typical and challenging chicken and egg problem because locating the robot's position depends on the coordinates of the surroundings and obtaining the coordinates of the surroundings also requires the robot's position [3]. Therefore, SLAM technology takes the form of building a map while localizing, where the robots travel through unknown environments through their own sensors such as odometers, LIDAR, cameras, RGB-D cameras, etc. At the same time, they can obtain their position and build a map of their surroundings.

Currently, the main SLAM technologies are Vision-SLAM (V-SLAM) and LIDAR SLAM. V-SLAM technology can combine the information captured by the camera with the map information to infer the location of the camera or the restaurant robot, and to construct a map of the local environment to which it currently belongs. The general process of V-SLAM is information acquisition, feature extraction, visual reconstruction, back-end optimization, and mapping. Information acquisition is the collection of data from cameras and Inertial Measurement Units (IMUs). Feature extraction is the process of combining images acquired by a series of cameras to find similar features. It is worth mentioning that if the camera is facing a featureless white wall, it will be difficult for it to complete feature extraction as well as perform localization and navigation [4]. The next step is visual reconstruction, which involves combining features and motion trajectories to reconstruct a 3D map [5]. Then comes the back-end optimization stage which refers to the optimization of camera trajectories and scene structure by minimizing the reprojection error. Currently, the mainstream back-end optimization methods are based on filtering theory such as the Extended Kalman Filter (EKF), optimization theory such as BA and graph optimization, and bitmap method. The last is to perform loopback detection and construct the map [6].

LIDAR SLAM is a method based on LIDAR to acquire information and complete SLAM. LIDAR SLAM is mainly divided into 2D and 3D LIDAR SLAM. LIDAR can be classified into single-line and multi-line LIDAR according to the number of LIDAR lines. Among them, 2D LIDAR SLAM mainly uses single-line LIDAR as the main sensor. LIDAR SLAM follows the process of information acquisition, front-end odometry, back-end optimization, loop detection, and mapping [7]. The front-end odometry part includes data preprocessing, point cloud registration, and pose estimation. The overall process of LIDAR SLAM is similar to visual SLAM, but unlike visual SLAM, LIDAR SLAM obtains point cloud data. Therefore, LIDAR SLAM typically extracts geometric features and completes data processing through point cloud registration for feature extraction. Also, due to the differences in the feature extraction step, LIDAR SLAM is more robust to various environments, while visual SLAM may be affected by factors such as changes in lighting and lack of texture. Because the main difference between LIDAR SLAM and visual SLAM lies in the front-end odometry part, next it will mainly introduce point cloud distortion correction and point cloud registration.

In order to integrate a navigation architecture for service robots and to facilitate the development of service robots, this paper analyzes the main process of SLAM which primarily applied to service robots in restaurant environments in real life, and mainly introduces the 3D feature reconstruction technology for SLAM.

2 3D Feature Reconstruction in Vision-SLAM

2.1 Information Acquisition

In V-SLAM, information is generally acquired by using cameras to take images of the environment. Depending on the mode of operation, cameras can be categorized into three types which are monocular cameras, stereo cameras, and RGB-D cameras. Monocular cameras have only one lens, so they provide images from a single viewing angle. Since there is only one viewing angle, monocular cameras do not have direct access to depth information. They are typically used for 2D image processing and motion estimation, and require other methods to estimate depth information, such as through motion vision or in combination with other sensors. Stereo cameras consist of two or more cameras distributed over a fixed baseline. Stereo cameras can provide depth information by simultaneously capturing different views of the scene. Stereo vision and depth estimation are achieved by comparing the image differences between the two cameras. RGB-D cameras combine the functionality of a normal color camera with that of a depth sensor. In addition to providing color images, RGB-D cameras are able to directly acquire depth information for each pixel. However, they have drawbacks such as narrow measuring range, small field of view, and susceptibility to sunlight interference [8].

2.2 Feature Extraction

Motion estimation and 3D reconstruction are very difficult due to the rich information and high computational complexity of some restaurant environment images. To facilitate computer recognition, it is necessary to extract some special regions called feature points in the image. Feature points usually consist of two parts. These two points are respectively the description of the feature points in the image and the description around the feature points [8]. After extracting the feature points, SLAM correlates the data. Then, the map as well as the position and attitude of the robot are reconstructed by the multi-view geometry theory.

Commonly used feature extraction and matching algorithms mainly include Scale-Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), Oriented FAST and Rotated BRIEF (ORB) and so on.

SIFT is a feature extraction algorithm based on scale invariance. It detects local extreme points in the image and extracts features on different scale spaces, making the feature points invariant to changes in scale, rotation and illumination. In the SIFT algorithm, the extreme values in the scale space are first detected and key points are located. Subsequently, the keypoint orientation is assigned and the keypoint descriptor is generated. Scale space extreme value detection utilizes Gaussian difference pyramids to detect extreme points in the image, which are considered as potential key point candidates. Key point localization determines the final key point location by pinpointing and filtering the extreme points. Key point direction assignment assigns a dominant direction to each key point, improving the rotational invariance of the feature. Key point descriptor generation utilizes the gradient information around the key points to construct feature descriptors that describe the local features around the key points. However, before the advent of GPUs, it could not fulfill the real-time requirements [9].

SURF is a method based on the FAST feature detector and Haar feature descriptors, aimed at extracting features. The algorithm mainly consists of three steps: scale-space extrema detection, key point localization, and feature descriptor generation. Scale-space extrema detection involves convolving the image with a box filter in the scale space to detect extrema points. Key point localization identifies extrema points in the Hessian matrix of the image and performs precise localization and filtering to determine the final key point positions. Feature descriptor generation constructs feature descriptors using Haar wavelet responses around key points to describe the local features of key points [10].

ORB is a feature extraction algorithm that combines FAST and BRIEF. It uses FAST algorithm to detect key points and then uses binary string feature descriptors. FAST corner detector is used to quickly detect key points in the image and BRIEF feature descriptor is used to quickly generate descriptors of these key points. Therefore, it has the advantages of SIFT and SURF algorithms which are faster and good robustness. ORB also introduces the concepts of oriented and rotated, which allows the algorithm to adapt to rotational transformations of the image [11].

Feature dependency has always been a significant limiting factor for V-SLAM. V-SLAM based on filters and optimization theory typically requires feature point extraction and matching in images to complete the V-SLAM process. Therefore, this method relies more on the quality and feature richness of the image. In contrast, Direct Tracking could directly solve camera motion by comparing pixel colors, thus generally exhibiting better robustness in cases of feature scarcity or image blur [4]. However, direct tracking introduces significant computational overhead, and running it on devices with lower computational performance may require reducing the resolution of sampled images, thereby compromising tracking and navigation accuracy [12]. Therefore, addressing feature dependency constraints on restaurant robots can enable them to be more widely applicable in various environments while maintaining efficiency.

2.3 Visual Reconstruction

The multi-view principle is used to convert 2D maps into 3D maps to complete the visual reconstruction. The visual reconstruction is accomplished by converting a 2D map into a 3D map using the principle of multiple viewpoints. The goal is to recover the camera motion parameter C_i and the 3D structure X_i of the scenes for each frame. where the motion parameter contains the camera position denoted by R_i and the orientation information denoted by P_i. R_i, P_i transforms the 3D point X_J in the global coordinate system to the local coordinate system of C_i.

$$(X_{ij}, Y_{ij}, Z_{ij})^T = R_i(X_j - p_i) \tag{1}$$

Then projected into the image:

$$h_{ij=}(f_x X_{ij}/Z_{ij} + c_{x,}\, f_x Y_{ij}/Z_{ij} + c_{y,})^T) \tag{2}$$

, where f_x, f_y are the image focal lengths along the image x, y axes, respectively. (c_x, c_y) is the position of the lens center of light in the image.

From equations (1)(2), it can be seen that the projected position of a 3D point h_{ij} in the image can be expressed as a function of C_i and X_j, denoted as

$$h_{ij} = h(C_i, X_j) \tag{3}$$

The next step required to match the same feature points in different images is known as feature matching. The following objective function is optimized by solving:

$$\underset{C_1 \ldots C_m,\, X_1 \ldots X_n}{\operatorname{argmin}} \sum_{i=1}^{m} \sum_{j=1}^{n} || \, h(C_i, X_j) - \widehat{x_{ij}} \, || \, _{\Sigma_{ij}} \tag{4}$$

Obtain a set of optimal $C_1 - C_m$, $X_1 - X_n$ such that the projected positions $h_{ij}$of all X_j in the C_i image are as close as possible to the observed image point positions x_{ij} [4].

For the binocular camera, the visual reconstruction process is similar to the monocular camera. However, since the binocular camera consists of two monocular cameras, the depth value of the object can be directly calculated by fusing the two acquired images. This overcomes the disadvantage that a monocular camera cannot determine distance from a single photograph at a single moment.

3 3D Feature Reconstruction in LIDAR SLAM

3.1 Information Acquisition

LIDAR SLAM performs information acquisition by carrying LIDAR. LIDAR is categorized into 2D LIDAR and 3D LIDAR. LIDAR SLAM selects the appropriate LIDAR according to the characteristics of different environments. Considering that some indoor restaurant environments are relatively simple and the functional requirements for the mobile robot are low, the mobile robot can choose 2D LIDAR. When the indoor restaurant environment factors are variable and more complex, the mobile robot uses 3D LIDAR to dynamically scan the three-dimensional space. LIDAR in the process of work, by scanning the environment information to obtain dispersed points, these points contain a variety of information, such dispersed points are called a point cloud when gathered [13]. After that, compare the information of each point to get the degree of change of distance and angle, and by calculating the difference of distance and angle, the change of position of the mobile robot can be obtained.

3.2 Point Cloud Distortion Correction

In LIDAR SLAM, point cloud distortion refers to the motion distortion of point cloud data. During the scanning process of the LIDAR, the robot carrying the LIDAR moves continuously, resulting in point cloud data in the same frame being measured from different positions of the LIDAR coordinate system. This distortion leads to increased positioning errors, making the system unstable. Common methods to remove point cloud motion distortion include pure estimation and sensor-assisted methods. In the pure estimation approach, the Iterative Closest Point (ICP) method is a classic technique that iteratively solves pose transformation using least squares. However, pure estimation methods fail to completely resolve motion distortion issues and instead employ techniques to mitigate their effects. Introducing external sensor-assisted methods is the current optimal solution. External sensor-assisted methods primarily utilize high-frequency sensors such as IMUs to directly measure angular velocity and linear velocity and compensate for motion in the point cloud.

3.3 Point Cloud Registration

Another significant function of the front-end odometry is to perform point cloud registration, calculating the poses between adjacent data frames. Registration algorithms mainly include ICP, NDT methods based on mathematical features, and learning-based methods. ICP is a classical registration algorithm that estimates the transformation between point clouds by minimizing the distance between them, typically used for matching dense point clouds.

Building upon the ICP algorithm, there are several variant algorithms such as Point-to-Plane ICP (PP-ICP), Point-to-Line ICP (PL-ICP), Normal ICP (NICP), and Implicit Moving Least Squares ICP (IMLS-ICP). PP-ICP replaces the traditional point-to-point distance calculation with point-to-plane distance calculation, leading to faster convergence speed, suitable for 3D LIDAR SLAM. PL-ICP improves matching accuracy by calculating the minimum distance between points and lines, applicable to both 2D and 3D LIDAR SLAM. NICP incorporates surface normal vector information into the ICP algorithm, enhancing registration accuracy and robustness. IMLS-ICP represents the point cloud surface using implicit functions and performs point cloud registration by minimizing the distance between MLS approximation and implicit functions, effectively handling registration problems of non-rigid objects.

Moreover, the Normalized Distribution Transform (NDT) method based on mathematical features models discrete point cloud data using Gaussian distribution, transforming point clouds into a point distribution function and calculating the relative pose relationship between point clouds. The NDT method has the advantages of fast speed, good stability, and high accuracy, suitable for matching sparse point clouds and non-rigid objects.

Traditional point cloud registration methods, despite achieving good results in some scenarios, still lack a universal solution applicable to all scenarios. Due to the rapid development in the field of deep learning, solutions based on deep learning for point cloud registration have become a new research direction for researchers. Emerging deep learning-based methods, such as Fully Convolutional Geometric Features (FCGF) and SpinNet, can extract richer point cloud features with rotational invariance. These deep learning-based methods can obtain more accurate models through data-driven learning but require a large amount of GPU resources for pre-training and model deployment [14].

4 Back-End Optimization and Mapping

Back-end optimization is used to improve the system's localization accuracy and map consistency by globally optimizing the robot trajectory and map. The mainstream back-end optimization methods currently applied to restaurant environments are filter theory-based methods, and optimization theory-based methods.

The use of back-end optimization methods based on filter theory is relatively uncommon in restaurant mobile robot applications. Although filter theory-based methods have an important role in real-time systems and sensor data processing, they are typically more suited to online, incremental state estimation and updating rather than global map optimization. Restaurant mobile robots typically need to navigate and operate in pre-built environments, and thus prefer to use graph-based optimization methods for back-end optimization. These methods enable global optimization of robot trajectories and maps to improve positioning accuracy and map consistency. Graph-based optimization methods are better able to handle large-scale maps and long-running systems, and can optimize in offline environments, making them more suitable for restaurant mobile robot scenarios. Therefore, the next section will focus on graph optimization-based algorithms. Then four common map representations will be briefly introduced.

4.1 Filter-Based Theory

Back-end optimization methods based on filter theory mainly use filters to estimate and update the state of the SLAM system. These methods typically include steps of state estimation, state prediction, observation updating and error optimization. The algorithm that is more commonly applied to robotic SLAM is the Extended Kalman Filter (EKF). State estimation is the estimation of the state of the system based on sensor observations and motion models through filters such as extended Kalman filter, extended information filter or particle filter. The state usually includes, for example, the robot's position and the location of feature points on the map. State prediction is the prediction of the system state at the next moment based on the motion model and control inputs, and the calculation of the state uncertainty. Then observation updating is to update the state based on the observations from the sensors using filters to fuse the observations and update the estimates and uncertainties of the state. Finally, error optimization is the periodic optimization of the system state to minimize the estimation error of the filter and to improve the accuracy and consistency of the system [15].

4.2 Optimization-Based Theory

Back-end optimization methods based on optimization theory refer to the optimization of state estimation and map construction for SLAM systems by minimizing the cost function of the system state. Optimization theory-based methods are graph optimization, least squares optimization and nonlinear optimization. Least squares optimization usually refers to the Bundle Adjustment (BA) algorithm. In the BA algorithm, the state variables of the system are adjusted by minimizing the sum of squares of the observed residuals to minimize the difference between the observed and predicted data. Nonlinear optimization is used to solve the SLAM problem using the Gauss-Newton and Levenberg-Marquardt methods. These methods are suitable for dealing with the nonlinear and non-Gaussian distribution characteristics of SLAM problems and can improve the stability and robustness of the system. The steps of the graph optimization methods that contribute more to mobile robots in restaurant environments are briefly described next.

The graph optimization algorithm begins by modeling the optimization problem as a graph, where nodes represent the state variables of the system such as the robot's position and the feature points of the map, and edges represent the constraints such as the sensor measurements and the motion model. Then a corresponding cost function is defined for each node and edge. The cost function describes the error of that variable or constraint. Usually, the sum of squares of the errors or some other form of error measure is used. An optimization algorithm is used to iteratively optimize the graph and attempt to adjust the state variables of the nodes to minimize the cost function. In each iteration, according to the results of the optimization algorithm, update the state variables of the nodes in the graph, so that it is gradually close to the optimal solution. Monitor the change of the cost function and set the convergence criterion to determine whether the optimization algorithm converges to the optimal solution. When the change in the cost function or the change in the state variables is less than a set threshold, the algorithm is considered to have converged and outputs the optimal values of the state variables which are used as the optimal solution of the problem. Through these steps, the graph optimization algorithm can globally optimize the state variables of the system, improve the accuracy and consistency of the SLAM system, and is able to handle complex nonlinear constraints and high-dimensional state spaces.

4.3 Mapping

3D maps can be expressed in four ways which are depth, point cloud, voxel, and mesh. In a depth map, each pixel represents the value of the object's distance from the camera plane. The main approaches are stereo vision and shadow shape algorithms, Hidden Markov Model modeling, Markov Random Field computation of parameters, and depth prediction algorithms based on deep learning. Point cloud as mentioned above in LIDAR SLAM is a dataset consisting of many points, each containing information such as position, color, etc. A point cloud map is a map composed of point clouds instead of pixels. The main methods for constructing point cloud maps are point cloud-based 3D reconstruction network, ElasticFusion algorithms, etc. Voxels are made up of small rectangular squares, similar to pixels in 3D space. The main methods to construct voxel maps are through the 3D-R2N2 algorithm, deep convolutional neural network combined with LSTM and GRU. A mesh is a polygon formed by stitching together many triangular faces. The advantage of using mesh is that it can be close to the surface of real objects. Its commonly used method is the Pixel2Mesh algorithm [6].

5 SLAM Challenges in Restaurant Environment

With the development of various technologies, SLAM has also gradually matured. However, there are still many challenges such as environmental complexity, accumulation of errors, cross-sensor fusion, fast motions, degenerate environments and sensor degradation, and robustness that need to be addressed in SLAM. This paper mainly focuses on accumulation of errors, restaurant robot's fast motions, degenerate environments, and sensor degradation.

5.1 Accumulation of Errors

Since SLAM systems measure and localize using relative sensors, these sensors can be affected by noise, light, and so on, which can lead to localization errors. When entering the motion estimation phase, the robot estimates its present pose based on its known recent past position. Once there is an error in the previously performed position estimation the subsequent relative position also exists generating further errors. This phenomenon of continuous superposition of errors is called error accumulation. It leads to uncertainty in trajectory estimation. Accumulated positioning errors may lead to map drift. While error accumulation can remain accurate in a small range of paths, it can lead to larger uncertainty when placed in a global map, thus affecting the accuracy of the robot's real-time localization and map building in the restaurant environment. Due to the dynamic nature of the restaurant environment, such as tables and chairs being moved or crowds moving, the robot must be able to correct these errors promptly to ensure the accuracy and reliability of the map. Therefore, appropriate algorithms and techniques need to be developed and employed to minimize the accumulation of errors, such as using closed-loop detection techniques or periodic calibration. Closed-loop detection is used to detect if the camera or robot has returned to a previously visited location, thus recognizing and correcting loops in the map, and reducing uncertainty by limiting it to previous poses with less accumulated drift. Another way to reduce drift is through periodic calibration, such as regular adjustment and calibration of the sensors used by the robot after it has been operating for a period to ensure the accuracy and consistency of the sensor measurements. This will eliminate the error at its root thus avoiding error accumulation. Therefore, drift reduction remains a major challenge and goal in SLAM algorithms.

5.2 Restaurant Robot's Fast Motions

The restaurant robot, serving as a service robot, should consider efficiency alongside accurate completion of meal delivery tasks. However, rapid and abrupt movements pose challenges for SLAM algorithms. In contrast, slow and smooth movements are easier to track. Besides motion blur or distortion, this may make feature extraction and matching more difficult. This is because fast movement causes the sensor to have fewer overlapping areas and features in photos taken at adjacent times, which is equivalent to a disguised reduction in the number of pictures in which the same features appear at the same period of time. Thus, the efficiency of feature matching and the accuracy of position estimation are reduced. What's more, in restaurant robot's fast motion, how to avoid moving pedestrians also needs to be considered. In a restaurant with a small space or when the dining period is peak, how to make the restaurant robot avoid moving pedestrians while maintaining high speed is a problem worth thinking about

5.3 Multi-Sensor Fusion

SLAM multi-sensor fusion faces multiple challenges. Firstly, the heterogeneity of different sensor types needs to be handled effectively to ensure data accuracy and consistency. Secondly, data alignment is necessary to address the temporal and spatial bias of different sensor data. The design of sensor fusion algorithms needs to consider factors such as correlation between sensors and information weight allocation. In addition, real-time and computational complexity are challenges that require minimizing the computational load while ensuring accuracy. Finally, environmental changes and external interference may affect the effectiveness of sensor fusion, so the system needs to have a certain degree of stability and robustness. Taking these factors into account, appropriate methods and techniques are needed to address the challenges of SLAM multi-sensor fusion.

6 Conclusion

This paper summarizes the flow of SLAM algorithms and in each flow introduces some of the classical methods commonly used in restaurant environments. Visual reconstruction techniques for V-SLAM and LiDAR SLAM are first highlighted. Then the commonly used back-end optimization techniques are introduced. Subsequently, the maps are categorized and the mapping methods for the corresponding categories are briefly summarized. Finally, the current challenges in the SLAM field and the future research directions of SLAM technology are analyzed. This paper provides methodological guidance for mobile robots working in restaurant environments. Combined with the path planning method for restaurant mobile robots, it can provide algorithmic ideas for the construction of restaurant mobile robots. In turn, it promotes the application of indoor mobile robots in restaurant environments. It improves the service quality and efficiency of service industries such as restaurants and reduces the cost. Thus, it promotes the rapid development of restaurants and other service industries combined with service robots.

References

1. Cord, T., Rupp, T., Lazic, D. E.: A navigation system for mobile service robots. In: Intelligent Autonomous Vehicles, pp. 225-230. IF AC, Madrid, Spain (1998).
2. Park, J.-H., Baeg, S.-H., Ryu, H.-S., Baeg, M.-H.: An intelligent navigation method for service robots in the smart environment. In: Proceedings of the 17th World Congress, pp. 1691-1696. IF AC, Seoul, Korea (2008).
3. Jia, Y., Yan, X., Xu, Y.: A Survey of simultaneous localization and mapping for robot. In: 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), pp. 857-861. Chengdu, China (2019).
4. Liu Haomin, Zhang Guofeng, Bao Hujun.: A Survey of Monocular Simultaneous Localization and Mapping. Journal of Computer-Aided Design & Computer Graphics 28(6), 855-868 (2016).
5. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. 2nd edn. Cambridge University Press, Canberra, Australia (2004).
6. Zhu, D., Xu, G., Zhou, J., Di, E., Li, M.: The Development of Visual SLAM Algorithms: A Survey. Communications Technology 54(3), 523-533 (2021).
7. Zhang, G., Yang, C., Wang, W., Li, Y.: A review of automatic driving simultaneous localization and mapping methods based on laser radar. Chinese Journal of Automotive Engineering 14(1), 1-13 (2024).
8. Chen, Y., Zhou, Y., Lv, Q., Deveerasetty, K. K.: A review of V-SLAM. In: Proceedings of the IEEE International Conference on Information and Automation, pp. 603-608. Wuyi Mountain, China (2018).
9. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60, 91–110 (2004) .
10. Bay H., Ess A., Tuytelaars T., et al.: Speeded-Up Robust Features (SURF). Computer Vision & Image Understanding 110(3), 346-359 (2008).
11. Rublee E., Rabaud V., Konolige K., Bradski G.: ORB: An efficient alternative to SIFT or SURF. In: International Conference on Computer Vision, pp. 2564-2571. Barcelona, Spain (2011).
12. Schöps T., Engel J., Cremers D.: Semi-dense visual odometry for AR on a smartphone. In: IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 145-150. Munich, Germany (2014).
13. Tian, Y., Chen, H., Wang, F., Chen, X.: Overview of SLAM Algorithms for Mobile Robots. Computer Science 48(9), 223-234 (2021).
14. Liu M., Xu G., Tang T., Qian X., Geng M.: Review of SLAM Based on Lidar. Computer Engineering and Applications 60(1), 1-14 (2024).
15. Holmes S. A., Klein G. and Murray D. W.: An O(N^2) Square Root Unscented Kalman Filter for Visual Simultaneous Localization and Mapping. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 7, pp. 1251-1263. (2009).