A State-of-the-Art Review on SLAM

Xuewei Zhou and Ruining Huang(✉)

School of Mechanical Engineering and Automation, Harbin Institute of Technology, Shenzhen 518055, China
hrn@hit.edu.cn

Abstract. SLAM (Simultaneous Localization and Mapping), also known as CML (Concurrent Mapping and Localization), refers to real-time positioning and map building, or concurrent mapping and positioning. After nearly 30 years of research on SLAM, there have been quite a few breakthroughs in the SLAM community. This paper aims to provide an insightful review of information background, recent development, feature, implementation, and recent issue in SLAM. This paper includes the following parts: First of all, it gives an overview of the basic development of SLAM from its introduction to the present. Then, and most importantly, it summarizes the mainstream SLAM technology and theoretical basis. In addition, some cutting-edge and novel SLAM research results are discussed respectively. Finally, this paper summarizes and introduces some practical applications of SLAM technology.

Keywords: Simultaneous localization and mapping · Perception · Robots · Sensing

1 Introduction

SLAM is the abbreviation of simultaneous positioning and mapping. It contains two main tasks: positioning and mapping. This is an important open problem in mobile robots: to move accurately, mobile robots must have an accurate environmental map. However, to build an accurate map, the mobile robot must accurately perceive the position [30].

In 1990, Randall Smith [43] first proposed to use EKF to incrementally estimate the posterior distribution of the robot's posture and the position of the landmark.

In 2006, Durrant-Whyte and Bailey proposed the term SLAM for the first time and determined a detailed probabilistic theoretical analysis framework for SLAM problems. Computational efficiency, data association, convergence, and consistency of SLAM are discussed [12]. SLAM has entered the era of systematic research.

With the rapid development of SLAM, in terms of sensors, SLAM, which has emerged in recent years, is equipped with LiDAR, camera, IMU, and other sensors [11]; in terms of the method of state estimation, SLAM's initial system based on filters (KF, EKF, PF) has gradually developed into a system based on

H. Liu et al. (Eds.): ICIRA 2022, LNAI 13457, pp. 240–251, 2022.
https://doi.org/10.1007/978-3-031-13835-5_22

graph optimization [8,29]; in terms of the perspective of algorithm architecture, the single thread has been replaced by multi-thread [28]; with the integration of multi-sensors, SLAM technology has changed from the earliest military prototype to the necessary application technology of robots today.

Many scholars have reviewed the evolution of SLAM, but most of them focus on a specific topic. Generally, researchers summarized the development process of SLAM so far into three main stages [7].

Classic Age (1986–2004). In the early stage, the definition of SLAM problem, modeling and solving method based on probability framework.

Algorithm-Analysis Age (2004–2015). In-depth study of some properties of SLAM problems, such as sparsity, convergence, consistency, and more diverse and efficient algorithms have been proposed one after another.

Robust-Perception Age (2015–). Start to consider the robustness, scalability, efficient algorithm under resource constraints, high-level semantic cognitive task orientation.

This paper gives a broad overview of the current state of SLAM research and offers the perspective of part of the community on the open problems and future directions for the SLAM study. Our main focus is on multi-sensor fusion, dynamic environment, and semantic SLAM, which are the hot spots that most scholars have studied and paid attention to in SLAM research. Then, some other potential research directions are summarized. Finally, we discuss the application scenarios of SLAM technology in real life.

2 Implementation of SLAM System

SLAM system mainly includes front-end and back-end. The front-end is responsible for extracting features and data association from sensor data, while the back-end is responsible for maximum posterior estimation, filtering, and closed-loop detection.

2.1 The Structure of SLAM System

The specific SLAM system can be regarded as the combination of sensor, map structure, and solution method, and different combination methods can produce different technical frameworks. Here are some commonly used sensors, map structures, and solutions in SLAM.

Sensors. Angle measuring sensors such as LiDAR, millimeter-wave radar, and sonar. Various cameras, such as RGB-D cameras, Stereo cameras, and Event cameras. Other sensors such as wheeled odometer, IMU, and GPS.

Map Structure. The structure is closely related to the sensor. When sensors measure distance and angle or extract feature points from vision, maps are usually based on landmarks. Grid map and points cloud map are also used in LiDAR SLAM. Besides, there are other types of maps such as geometric maps which are based on edge or surface, octree maps [23], and so on.

Solution Method. Bayesian-framework-based recursive filtering, such as EKF, PF, and IF. Optimization-based method, one is the Bundle Adjustment [34]; The other is graph optimization [29].

2.2 LiDAR SLAM System

In the early stage of SLAM development, the complete SLAM was basically based on LiDAR. The data acquired by the laser is the depth value with angle, which can also be regarded as a points cloud. The rigid body motion of laser radar at two adjacent moments can be obtained by inter-frame matching of the point clouds scanned by laser radar. The most important and classic algorithm is the ICP [4], which is not limited to laser SLAM but is widely used in SLAM problems.

It is worth noting that it takes time for LiDAR to acquire complete data, which will cause each laser spot to be generated at a different reference pose in the process of robot movement. When the scanning frequency of LiDAR is relatively low, the error caused by robot motion can't be ignored. engineers can use the pure state estimation method, such as the variant of the ICP: VICP [22], to remove the motion distortion, or use sensor-assisted methods, such as odometer and IMU.

Early LiDAR SLAM algorithms did not include loop closing detection. Later, Scan Context [26] was used for fast inter-frame matching to achieve closed-loop detection. Recently, the paper [9] has proposed a state-of-the-art closed-loop detection method, which uses Deep Learning to search the loop closing based on different clues from LiDAR data. This method is more effective and demonstrates the great potential of deep learning.

SLAM based on LiDAR is the most mature and commonly used solution in business. Classical LiDAR SLAM systems include Gmaping [19], Catigrapher [21], and LOAM-SLAM [25], etc. LiDAR SLAM in indoor environment has a very good practical effect.

2.3 Visual SLAM System

Different from LiDAR, a vision sensor has a lot of information, which can provide us with a large amount of data. The cameras used in visual SLAM can be divided into three categories according to their working modes: monocular camera, stereo camera, and depth camera (RGB-D). Among them, the monocular vSLAM has scale uncertainty in theory, because the distance of the object is lost in the projection process of the monocular camera [36].

The practice shows that VO has the most significant influence on SLAM [40]. According to the map structure, VO can be divided into dense and sparse; according to the form of error, VO can be divided into two categories: indirect method, also known as feature-base method, and direct method.

Indirect method means that by detecting the feature points of an image, calculating the descriptors, and then matching the feature points of two adjacent images, the corresponding relationship between the feature points of two images can be obtained. When there are enough corresponding feature points, we can use geometric constraints [20] to solve the relative pose of two adjacent frames and get the 3D coordinates of feature points. RANSAC is used [13] to remove the unreliable matching point pairs.

The direct method is directly evolved from the Optical Flow method [3], and has the same assumption. The method directly estimates the motion of the camera according to the brightness information of the pixels in two adjacent frames of the camera. Image Pyramid [1] is introduced to improve the convergence of the algorithm.

As result, Map points are more parameterized by inverse depth [10], and vSLAM is more inclined to be used in conjunction with IMU [42]. Bag of words model [17] is often used in closed-loop detection and feature matching.

3 Current Mainstream SLAM

The state-of-the-art SLAM is divided into three main research directions, including multi-sensor fusion, using semantic information, and ensuring the robustness of the SLAM system in a dynamic environment. These three directions also overlap each other.

3.1 Multi-sensor Fusion

In multi-sensor fusion, it is inevitable to calibrate the external parameters of sensors, and the fusion algorithms can basically be classified into two categories: filter and graph optimization. This section will introduce VIO and LVIO, and the research status of multi-sensor fusion.

Visual Inertial Odometry. IMU has a high sampling frequency, is not affected by the external environment, and can estimate the absolute scale. At the same time, it can use the information of visual positioning to estimate the bias of IMU, which is complementary to the visual positioning scheme. Therefore, there are quite a few achievements in recent years [24]. How to integrate vision with IMU can be divided into two schemes: loose coupling and tight coupling.

Loose coupling refers to the direct fusion of IMU positioning and vision /GNSS pose. The fusion process does not influence the initial systems, and it is output as a post-processing mode. A typical example is the Kalman filter. The tightly coupled fusion process will affect the parameters of vision and IMU. The current open-source packages are basically based on tight coupling because tight coupling can model all the motion and measurement information at once, and it is easier to get the best estimation.

IMU's pre-integration was first put forward by Lupton in 2012 [35], and further extended to Lie algebra by Forster in 2015 [14,15]. This set of pre-integration theories has been widely used in VIO based on the BA optimization framework, which can greatly reduce the calculation amount of the algorithm and ensure the real-time performance of the algorithm.

Optimization-Based Example: VINS-Mono [48] is a well-known monocular VIO algorithm that was opened in 2018 by the Flying Robot Laboratory of Hong Kong University of Science and Technology. While ensuring the high-precision odometer effect, it can also estimate the sensor external parameters, IMU bias, and sensor time delay at the same time. It is a very classic and excellent VIO framework.

Filter-Based Example: MSCKF [38] stands for Multi-State Constraint Kalman Filter, which is a VIO algorithm based on filtering. It was first proposed by Mourikis in 2007. MSCKF fuses IMU and visual information under the framework of EKF. MSCKF is widely used in robots, UAVs, and AR/VR. For example, Google Project Tango uses MSCKF for pose estimation.

Lidar-Visual-Inertial Odometry. There are three ways to fusion LiDAR and vision. Use visual information to improve the accuracy of LiDAR, use LiDAR information to improve the accuracy of vision, and use both LiDAR and visual information [11].

3.2 Using Semantic Information

The research work of the traditional SLAM algorithm is faced with relatively simple geometric features such as points, lines, and surfaces, and almost no object-level features are involved. Secondly, the map constructed by the traditional SLAM algorithm cannot be "reused" in practice and lacks the visual map that can be directly used in actual production.

The original purpose of using Semantic information in SLAM is to match object-level features and build a map with semantic information. It has made great progress under the impetus of deep learning and has become a relatively independent branch. There are two main aspects of the combination of SLAM and semantics.

One is semantics help SLAM. Information about objects can help to get maps with object labels, which are easier to be understood by human thinking. In addition, the tag information of the object can bring more constraints for loop detection and BA optimization.

The other one is SLAM help semantics. In SLAM, engineers can collect data on objects from different perspectives and estimate their pose, automatically generate high-quality sample data for semantic recognition, avoid manual calibration of data, and speed up the training process of classifiers.

The concept of semantic SLAM is vague. At present, the so-called semantic segmentation based on neural networks, object detection, instance segmentation, and other technologies are used in SLAM, mostly for feature point selection and camera pose estimation. More broadly, the methods that use neural networks such as end-to-end image pose, marking point cloud from segmentation results, scene recognition, feature extraction, and loop detection can be called Semantic SLAM.

Semantic SLAM-related work involves a wide range of aspects, mainly including feature selection [18,31], dynamic scene [6,49,51], monocular scale restoration [16,45], long-term positioning [44] and improving algorithm accuracy [5,32].

3.3 Dynamic Environment

The main problem in dynamic SLAM is handling dynamic data associations. By choosing whether to cull dynamic correspondences or use them to track objects, the dynamic SLAM problem can be considered a robustness problem or an extension of standard SLAM [39]. To understand the dynamic environment SLAM problem, developers must first understand what disadvantages the dynamic environment has brought to SLAM so that researchers need to solve it.

The first is the SLAM registration level. No matter which point cloud registration method, it is based on static assumption. In theory, dynamic points will definitely affect the registration accuracy. At this level, dynamic points can only be identified and killed in real-time before or during registration. As for how to identify? Traditional methods, such as eliminating points that are too far away in the process of registration iteration, the more popular method at present is to directly identify and kill dynamic objects based on deep learning.

Secondly, the level of map building. It is assumed that the interference of dynamic objects to the registration is limited, which does not affect the trajectory accuracy, but it is still unbearable that the final generated map is full of ghosts of a large number of dynamic objects, which will have an adverse impact on map-based positioning or map-based feasible area planning.

Target Classification. At present, the main solution is to divide all objects in the environment into four categories according to their dynamic degree:

a) High dynamic objects;
b) Low-dynamic objects;
c) Semi-static objects;
d) Static objects.

Except for static objects, the other three types of objects all have dynamic attributes to different degrees, and their coping strategies are different. For high dynamic objects: online real-time filtering; For low dynamic objects: after a SLAM process, post-processing filtering; For semi-static objects: life-long mapping (or long-term mapping). The above three ways are upward compatible. No matter which one of these three coping ways, more papers have been published, and the selective excerpts are as follows.

Online Real-Time Filtering. Dynamic filtering must require reference frames to compare dynamic points. To achieve real-time, there will not be too many reference frames.

Dynamic Filtering in Post-processing Mode. As the post-processing method does not need to worry about real-time, all frames in the whole SLAM period can be used as reference information to identify dynamic points. Compared with the real-time method, the post-processing method pursues the accuracy and sufficiency of dynamic point cloud filtering. On the premise of post-processing, common dynamic object filtering methods can be divided into three typical categories: segmentation-based, ray-casting-based [41], and visibility-based [27].

Life-Long Mapping. In fact, the core problem of life-long mapping is far more than dynamic/semi-static object filtering. Dynamic/semi-static object filtering is only a part of map fusion among different sessions in the process of life-long, and map fusion is only a part of life-long mapping [50].

4 The Future of SLAM

The front part is just an overview of several main research directions of current SLAM, and there are many other aspects with the same research potential. In some classical SLAM theories, there are also outstanding works like ICE-BA [33].

4.1 Implicit Map Representation

Using a neural network to implicitly store the information of a 3D space, instead of using vectors or sets to explicitly store feature points or mesh as in the traditional SLAM, is a new technical route to solve old problems, which has a promising research prospect and has become a research hotspot since 2020. It is equivalent to putting cartography into the network, so there is no need for dense point clouds to save the map, and only small network parameters are needed to restore the scene.

iMAP. Drawing on the idea of NeRF [37], iMAP [46] in Andrew Davison's lab first proposed to use MLP (Multilayer Perceptron, a feedforward artificial neural network model) to represent the scene map in the SLAM process. This implicit map representation method can not only solve the problem of map storage but also control the details of scene reconstruction, such as the reconstruction of objects that cannot be observed by the camera.

NICE-SLAM. In this paper [52], the author puts forward NICE-SLAM, which is an intensive SLAM system, which combines multi-level local information by introducing hierarchical scene representation. By optimizing this representation with pre-trained geometric prior, details can be reconstructed on large indoor scenes. Compared with the recent neural implicit SLAM system, this method is more scalable, efficient, and robust.

4.2 Event Camera

The event camera is a new type of sensor, which outputs the change of pixel brightness. The amount of data in the event stream is much smaller than the data transmitted by the traditional camera, and the event stream has no minimum time unit, so it has low delay characteristics, unlike the traditional camera which outputs data regularly. The overall method framework of the camera is basically the same as that of the traditional camera field, that is, data association, observation model, residual calculation, and problem-solving. To sum up, the following three issues need to be considered in designing the event camera SLAM.

1) How to design a method directly based on event camera data, and meet the requirement of low computation.
2) How to find some kind of information, which is used to establish the data association of events.
3) Whether to use a monocular event camera or a binocular event camera.

4.3 Active SLAM

Active SLAM is different from the general SLAM method in that: the general SLAM method is to give the input data, and get the location and map by analyzing and calculating the input data; Active SLAM combines SLAM with path planning to explore the environment by planning the path of mobile robots, which makes SLAM algorithm get high-precision maps faster and better. The problem of controlling robot motion to minimize the uncertainty of its map representation and positioning is usually called active SLAM. This definition is derived from the well-known Bajcsy's active perception [2] and Thrun's robot exploration [47].

Active SLAM is a decision problem, and several general decision frameworks can be used as the backbone of exploration-development decisions.

A complete framework for active SLAM can be divided into three main steps [7]:

1) The robot identifies possible locations to explore or exploit in its current estimation of the map, i.e. vantage points.
2) The robot calculates the effect of visiting each vantage point and selects the action with the highest effect.

3) The robot performs the selected action and decides whether to continue or terminate the task.

For active SLAM to have an impact in practical applications, there are still many problems to be solved, including fast and accurate prediction of future states and convergence of mathematical optimal solutions.

5 The Application of SLAM

At present, SLAM technology is mainly used in UAVs, unmanned driving, robot, AR, smart home, and other fields, starting from various application scenarios to promote consumption upgrades.

5.1 Autonomous Vehicles

Unmanned driving is one of the hot topics in recent years, Google, Uber, Baidu, and other enterprises are accelerating the research and development of driverless technology, to seize the initiative. With the improvement of the Internet of things and intelligent systems in cities, driverless driving is bound to be the general trend. Unmanned vehicles use LiDAR sensors (Velodyne, IBEO, etc.) as tools to obtain map data and construct maps to avoid obstacles and achieve path planning. Similar to the application of SLAM technology in the field of robotics, but compared with the application of SLAM in robotics, unmanned radar requirements and costs are significantly higher than robots.

5.2 UAV

UAVs need to know where there are obstacles, how to avoid them, and how to re-plan their route during flight. Obviously, this is the application of SLAM technology. However, the UAV flight range is large, so the accuracy of the requirements are not high, some other optical flow, and ultrasonic sensors on the market can be used as auxiliary.

5.3 AR

AR applies virtual information to the real world through computer technology, and the real environment and virtual objects are superimposed into the same picture or space in real-time. The realization of this picture is inseparable from the real-time positioning of SLAM technology. Although there are many alternative technologies in the AR industry, SLAM is the most ideal positioning and navigation technology. Compared with the application of LAM in robotics, unmanned driving and other fields, the application of LAM in the AR industry has many differences.

6 Conclusion

Multi-sensor fusion, optimization of data association and loopback detection, integration with front-end heterogeneous processor, enhancement of robustness, and repositioning accuracy are the next development directions of SLAM technology, but these will be gradually solved with the development of consumption stimulation and industrial chain. Just like in mobile phones, SLAM technology will enter People's Daily life in the near future and change people's lives.

Acknowledgment. This work was funded by the National Natural Science Foundation of China (No. 51975155), the Natural Science Foundation of Guangdong Province (No. 2021A1515011823) and the Shenzhen Basic Research Program (No. JCYJ20200824082533001).

References

1. Adelson, E.H., Anderson, C.H., Bergen, J.R., Burt, P.J., Ogden, J.M.: Pyramid methods in image processing. RCA Eng. **29**(6), 33–41 (1983)
2. Bajcsy, R.: Active perception. Proc. IEEE **76**(8), 966–1005 (1988)
3. Baker, S., Matthews, I.: Lucas-Kanade 20 years on: a unifying framework. Int. J. Comput. Vision **56**(3), 221–255 (2004)
4. Besl, P.J., Mckay, H.D.: A method for registration of 3-D shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14**(2), 239–256 (1992)
5. Bowman, S.L., Atanasov, N., Daniilidis, K., Pappas, G.J.: Probabilistic data association for semantic slam. In: 2017 IEEE International Conference on Robotics and Automation (ICRA) (2017)
6. Brasch, N., Bozic, A., Lallemand, J., Tombari, F.: Semantic monocular slam for highly dynamic environments. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2018)
7. Cadena, C., et al.: Past, present, and future of simultaneous localization and mapping: toward the robust-perception age. IEEE Trans. Rob. **32**(6), 1309–1332 (2016)
8. Castellanos, J.A., Neira, J., Tardos, J.D.: Limits to the consistency of EKF-based slam. In: Symposium on Intelligent Autonomous Vehicles (2004)
9. Chen, X., Läbe, T., Milioto, A., Röhling, T., Behley, J., Stachniss, C.: OverlapNet: a siamese network for computing LiDAR scan similarity with applications to loop closing and localization. Auton. Robots **46**, 61–81 (2021). https://doi.org/10.1007/s10514-021-09999-0
10. Civera, J., Davison, A.J., Montiel, J.: Inverse depth parametrization for monocular SLAM. IEEE Trans. Rob. **24**(5), 932–945 (2008)
11. Debeunne, C., Vivet, D.: A review of visual-lidar fusion based simultaneous localization and mapping. Sensors **20**(7), 2068 (2020)
12. Durrant-Whyte, H., Bailey, T.: Simultaneous localisation and mapping (SLAM): part 2. IEEE Robot. Autom. Mag. **13**(3), 108–117 (2006)
13. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: Readings in Computer Vision, pp. 726–740 (1987)
14. Forster, C., Carlone, L., Dellaert, F., Scaramuzza, D.: IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation (supplementary material). Georgia Institute of Technology (2015)

15. Forster, C., Carlone, L., Dellaert, F., Scaramuzza, D.: On-manifold preintegration theory for fast and accurate visual-inertial navigation. Comput. Sci. (2015)
16. Frost, D., Prisacariu, V., Murray, D.: Recovering stable scale in monocular slam using object-supplemented bundle adjustment. IEEE Trans. Rob. **34**, 736–747 (2018)
17. Galvez-Lpez, D., Tardos, J.D.: Bags of binary words for fast place recognition in image sequences. IEEE Trans. Rob. **28**(5), 1188–1197 (2012)
18. Ganti, P., Waslander, S.L.: Visual slam with network uncertainty informed feature selection (2018)
19. Grisetti, G., Stachniss, C., Burgard, W.: Improved techniques for grid mapping with rao-blackwellized particle filters. IEEE Trans. Rob. **23**(1), 34–46 (2007)
20. Hartley, R.I., Zisserman, A.: Multi-view geometry in computer vision. Kybernetes **30**(9/10), 1865–1872 (2019)
21. Hess, W., Kohler, D., Rapp, H., Andor, D.: Real-time loop closure in 2D LIDAR SLAM. In: 2016 IEEE International Conference on Robotics and Automation (ICRA) (2016)
22. Hong, S., Ko, H., Kim, J.: VICP: velocity updating iterative closest point algorithm. In: IEEE International Conference on Robotics and Automation (2012)
23. Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W.: OctoMap: an efficient probabilistic 3D mapping framework based on octrees. Auton. Robot. **34**(3), 189–206 (2013)
24. Huang, G.: Visual-inertial navigation: a concise review. In: 2019 International Conference on Robotics and Automation (ICRA) (2019)
25. Zhang, J., Singh, S.: Low-drift and real-time lidar odometry and mapping. Auton. Robot. **41**(2), 401–416 (2016). https://doi.org/10.1007/s10514-016-9548-2
26. Kim, G., Kim, A.: Scan context: egocentric spatial descriptor for place recognition within 3D point cloud map. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2018)
27. Kim, G., Kim, A.: Remove, then revert: static point cloud map construction using multiresolution range images. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2020)
28. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: IEEE and ACM International Symposium on Mixed and Augmented Reality (2008)
29. Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W.: G2o: a general framework for graph optimization. In: IEEE International Conference on Robotics and Automation (2011)
30. Leonard, J.J., Durrant-Whyte, H.F.: Simultaneous map building and localization for an autonomous mobile robot. In: Proceedings IROS 1991. IEEE/RSJ International Workshop on Intelligent Robots and Systems 1991. Intelligence for Mechanical Systems (1991)
31. Liang, H.J., Sanket, N.J., Fermüller, C., Aloimonos, Y.: SalientDSO: bringing attention to direct sparse odometry. IEEE Trans. Autom. Sci. Eng. **16**(4), 1619–1626 (2018)
32. Lianos, K.N., Schnberger, J.L., Pollefeys, M., Sattler, T.: VSO: visual semantic odometry. In: European Conference on Computer Vision (ECCV) (2018)
33. Liu, H., Chen, M., Zhang, G., Bao, H., Bao, Y.: ICE-BA: incremental, consistent and efficient bundle adjustment for visual-inertial slam. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
34. Lourakis, M.: SBA: a software package for generic sparse bundle adjustment. ACM Trans. Math. Softw. **36**(1), 2 (2009)

35. Lupton, T., Sukkarieh, S.: Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. IEEE Press (2012)
36. Martinezmontiel, J.: Scale drift-aware large scale monocular SLAM. In: Robotics: Science and Systems VI, Universidad de Zaragoza, Zaragoza, Spain, 27–30 June 2010 (2010)
37. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ren, N.: NeRF: representing scenes as neural radiance fields for view synthesis. Commun. ACM **65**(1), 99–106 (2022)
38. Mourikis, A.I., Roumeliotis, S.I.: A multi-state constraint Kalman filter for vision-aided inertial navigation. In: 2007 IEEE International Conference on Robotics and Automation (2007)
39. Saputra, M.R.U., Markham, A., Trigoni, N.: Visual slam and structure from motion in dynamic environments: a survey. ACM Comput. Surv. (CSUR) **51**(2), 1–36 (2018)
40. Scaramuzza, D., Fraundorfer, F.: Visual odometry [tutorial]. IEEE Robot. Autom. Mag. **18**(4), 80–92 (2011)
41. Schauer, J., Nuechter, A.: The peopleremover-removing dynamic objects from 3-D point cloud data by traversing a voxel occupancy grid. IEEE Robot. Autom. Lett. **3**(3), 1679–1686 (2018)
42. Serviéres, M., Renaudin, V., Dupuis, A., Antigny, N.: Visual and visual-inertial slam: state of the art, classification, and experimental benchmarking. J. Sens. **2021**(1), 1–26 (2021)
43. Smith, R., Self, M., Cheeseman, P.: Estimating uncertain spatial relationships in robotics, pp. 435–461 (1988)
44. Stenborg, E., Toft, C., Hammarstrand, L.: Long-term visual localization using semantically segmented images, pp. 6484–6490 (2018)
45. Sucar, E., Hayet, J.B.: Bayesian scale estimation for monocular slam based on generic object detection for correcting scale drift (2017)
46. Sucar, E., Liu, S., Ortiz, J., Davison, A.J.: iMAP: implicit mapping and positioning in real-time (2021)
47. Thrun, S.: Probabilistic robotics. Commun. ACM **45**(3), 52–57 (2005)
48. Qin, T., Li, P., Shen, S.: VINS-mono: a robust and versatile monocular visual-inertial state estimator. IEEE Trans. Robot. **34**(4), 1004–1020 (2018)
49. Yu, C., et al.: DS-SLAM: a semantic visual slam towards dynamic environments. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2018)
50. Zhao, M., et al.: A general framework for lifelong localization and mapping in changing environment. arXiv e-prints (2021)
51. Zhong, F., Sheng, W., Zhang, Z., Chen, C., Wang, Y.: Detect-slam: making object detection and slam mutually beneficial. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV) (2018)
52. Zhu, Z., et al.: NICE-SLAM: neural implicit scalable encoding for slam (2021)