Research on Semantic Vision SLAM Towards Dynamic Environment

Nanyang Bai(✉), Tianji Ma, Wentao Shi, and Lutao Wang

Chengdu University of Information Technology, Chengdu, China
wanglt@cuit.edu.cn

Abstract. Simultaneous localization and mapping (SLAM) is considered to be the basic ability of intelligent mobile robots. In the past few decades, thanks to community's continuous and in-depth research on SLAM algorithms, the current SLAM algorithms have achieved good performance. But there are still some problems. For example, most SLAM algorithms have the assumption of a static environment, but in real life, most of the environment contains moving objects, so how to deal with the moving objects in the environment requires careful consideration. What's more, traditional geometric maps cannot specific environmental semantic information for mobile robots, so how to make robots truly understand the surrounding environment to complete some advanced tasks is also a difficult problem. In this paper, we design a scheme to improve the accuracy and robustness of SLAM in a dynamic environment. And we realize the perception of semantic information of objects in the environment through the object detection algorithm of deep learning neural network.

Keywords: SLAM · Semantic recognition · Semantic map · Dynamic target detection

1 Introduction

In the past years, with the development of AR (Augments Reality), UAV (Unmanned Aerial Vehicle), and UGV (Unmanned Ground Vehicle), visual SLAM has been extensively investigated. The mainstream vision sensors are divided into monocular cameras, stereo cameras, and RGB-D cameras. The monocular camera's simple solution has advantages in terms of size, power, and cost. However, there are also some problems, such as the inability to observe the scale and state initialization. By using more complex equipment, such as stereo cameras or RGB-D cameras, these problems could be solved, and the robustness of the visual SLAM system is also greatly improved.

Thanks to the SLAM system's continuous research by the research group, the visual SLAM system framework has been quite mature. It usually consists of several essential parts, such as feature extraction front-end, state estimation back-end, and closed-loop detection. Additionally, some advanced SLAM algorithms have achieved satisfactory performance, such as ORBSLAM2 [1], LSD-SLAM [2].

X. Wu et al. (Eds.): QShine 2020, LNICST 381, pp. 91–102, 2021.
https://doi.org/10.1007/978-3-030-77569-8_7

However, some issues remain unsolved. For example, these algorithms all assume the strong constraint of the static environment. When there are dynamic objects in the environment, its robustness and accuracy will be significantly decreased. Besides, these algorithms only provide geometric maps. It cannot provide support for advanced tasks such as intelligent obstacle avoidance.

In a dynamic environment, the SLAM algorithm's robustness will be significantly affected whether those algorithms are based on the feature method or the direct method because dynamic objects in the environment will corrupt the state estimation quality and lead to system failure. For example, dynamic objects in the environment may deceive the feature association in the visual SLAM algorithm. Therefore, to improve the entire system's stability, it is especially important to deal with dynamic objects in the environment.

The typical SLAM method only provides a geometric map composed of points and planes, which does not contain the surrounding environment's semantic information. Compared with geometric maps, semantic maps have the advantages of intuitive visualization and effective human–machine-environments interaction. According to the summary by Cadena et al. [3] We have now entered the third stage of SLAM research, videlicet, a stage of robust perception: the realization of robust performance, high-level understanding, resource perception, and task-driven perception represents the theme of this era.

This paper focuses on reducing the impact of dynamic objects in vision SLAM by combining the Mask R-CNN network with epipolar geometry. Simultaneously, the semantic information is bound to the octree map to obtain the semantic graph. Provide conditions for robots to achieve advanced tasks such as intelligent navigation.

In the rest of this paper, the structure is as follows. Section 2 provides an overview of semantic SLAM and SLAM in dynamic environments. Then Sect. 3 presents the framework of this whole SLAM system in detail. And we discuss how to detect dynamic objects and produce semantic maps. Subsequently, Sect. 4 provides the results of our program and ORB-SLAM2 on the TUM RGB-D dataset. Finally, in Sect. 5, we give a brief conclusion and discussion about our work.

2 Related Work

2.1 Semantic SLAM

Generally speaking, semantic SLAM is to use a neural network to provide road sign information for traditional SLAM solutions. The semantic SLAM system consists of two essential components: a semantic extractor and a modern V-SLAM framework. The semantic information is mainly extracted and derived from two processes. They are object detection and semantic segmentation [4].

Object detection is recognized as an essential branch of CV, whose development can be roughly divided into handcraft feature-based machine learning stage (2001–2013) and learning feature-based deep learning (2013-present). The former is extremely dependent on handcraft features of images [5–9]. It also requires many computing resources. In recent years, due to the introduction of deep learning and graphics processing units,

Object detection's accuracy and efficiency have been greatly improved in theory and practice. Therefore, more and more SLAM adds semantic modules into the system.

The earliest semantic map was proposed by Pham et al. [10] They used SSD, which has a fast detection speed, and ORB-SLAM2, which can achieve real-time positioning and promote each other. Then through dividing the depth map, object detection, and finally, output a semantic map with semantic information. Pronobis et al. [11, 12] proposed an online system using lasers and cameras to construct a semantic map of the environment. McCormac et al. proposed a dense three-dimensional semantic mapping method SemanticFusion [13] using convolutional neural networks. By combining CNNs with a dense SLAM solution ElasticFusion [14]. It ensures the dense long-term consistency of indoor positioning and eliminates the multi-circle scanning trajectory's cumulative error. It integrates the semantic prediction probability of CNNs from the multi-view points into the map to obtain a three-dimensional dense semantic map.

2.2 Dynamic Segmentation

In the SLAM community, relevant information extracted from static objects is considered stable and effective, while information extracted from moving objects is known to decline the algorithm's performance. For dynamic objects in the environment, advanced SLAM systems either treat them as outliers and eliminate them in different ways. Either use a separate target tracking module to track it.

One of the earliest works about SLAM in dynamic environments is presented by Hahnel et al. [15] use an Expectation–Maximisation (EM) algorithm to update the probabilistic estimate about which measurements correspond to a static/dynamic object and remove them from the estimation when they correspond to a dynamic object. Alcantarilla et al. [16] introduce dense scene flow for dynamic objects detection and show improved localization and mapping results by removing "erroneous" measurements on dynamic objects from the estimation. Tan et al. [17] propose an online keyframe update that reliably detects changed features in terms of appearance and structure and discards them if necessary. Kundu et al. [18] extend egomotion estimation with MBSfM [19] techniques similar to estimate the SE (3) trajectories of the third-party motions in a scene, but they constrain all the motions to the horizontal plane.

3 System Description

3.1 SLAM

In practical applications, the accuracy of attitude estimation and harsh environments' reliability are the critical factors for evaluating autonomous robots. ORB-SLAM2, as a relatively lightweight SLAM system, has an excellent performance in a static environment. We added a dynamic object detection module and a semantic map module for it. As shown in Fig. 1, we have designed five threads to control the SLAM system's five main modules. For the part of the dynamic environment, we place a real-time semantic segmentation network in a separate sub-thread and filter out the scene's dynamic targets by combining semantic segmentation and moving consistency checking methods. This

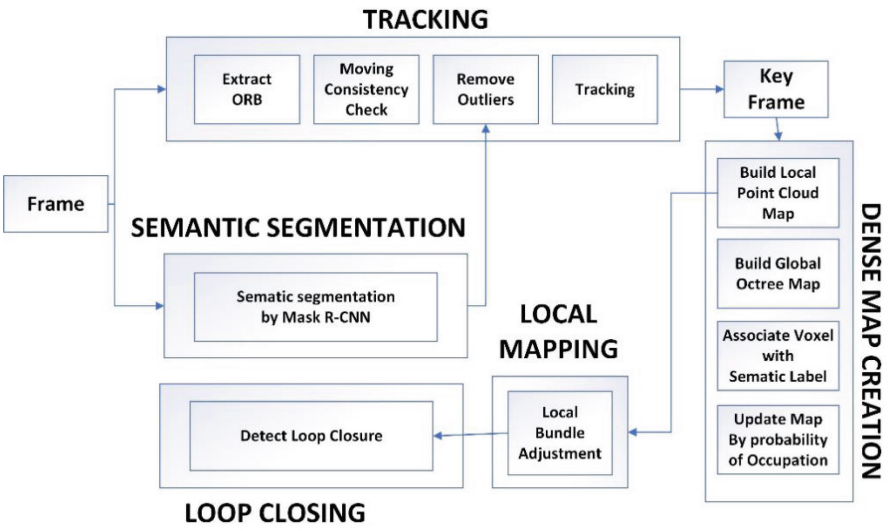

Fig. 1. The framework of our system. Our work mainly focuses on semantic extraction and dynamic detection of input images.

improves the robustness and accuracy of the SLAM system in dynamic scenarios. For the semantic map part, we also designed a separate thread to build the octree map. The semantic map is realized in the form of binding with semantic tags.

3.2 Semantic Segmentation

Semantic segmentation is an integral part of image processing and image understanding in machine vision technology. Semantic segmentation is to classify each pixel in the image, determine each point's category (such as background, person, or car), and divide the area.

To obtain the semantic information and potential moving objects in the environment, we use Mask R-CNN [20] to perform semantic segmentation on the image to obtain the objects in the environment and their semantic information. Mask R-CNN is a region-based semantic segmentation method that uses selective search to extract many target proposals. It then calculates the CNN features of each proposal. Finally, class-specific linear support vector machines are used to classify each region. Compared with other CNN networks, Mask R-CNN has higher speed and accuracy.

Like Fig. 2, it can subdivide the dynamic or movable objects in the image (such as people, animals, bicycles, cars, motorcycles, planes, buses, trains, trucks, boats, etc.). Most of the objects in our lives are included in their recognizable range.

3.3 Moving Consistency Check

In the Mask R-CNN network, we detect and process potential moving objects in the image, but these potential moving objects are not necessarily in a real state of motion.

Fig. 2. The result of object detection by Mask R-CNN in COCO dataset

They may also be static, such as a car parked on the side of the road. If we remove all the features of the image located on the potential moving objects, although the negative impact of motion on the accuracy can be avoided, we will also lose a large number of effective features. This may cause the tracking of the SLAM system to be lost due to the lack of matching features. Therefore, it is particularly important to use the movement consistency to detect the true state of all potential dynamic objects. Geometric constraints (such as epipolar lines, triangulation, basic matrix estimation or reprojection error equations) are effective ways to determine the state of feature motion [21].

In this experiment, we use the feature of epipolar constraint to distinguish the dynamic and static features of the object. As shown in Fig. 3, the static feature satisfies the standard constraints of the epipolar geometry (Fig. 3 (a)). If the tracked feature is too far from the polar line, it is likely to be a dynamic feature (Fig. 3 (b)).

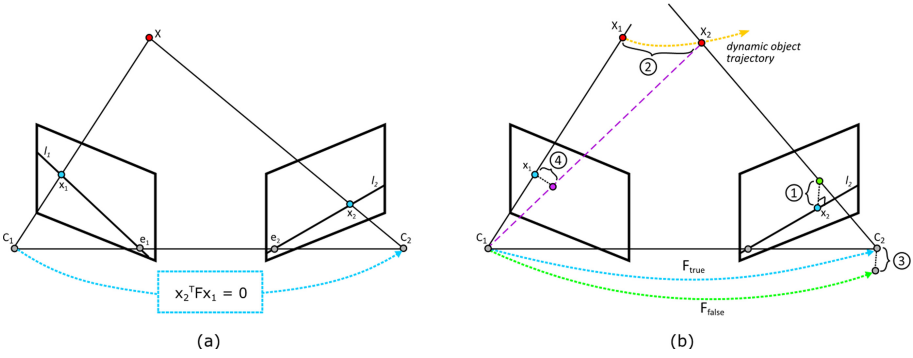

Fig. 3. (a) The transformation form point x_1 to point x_2 be define by epipolar constraint in static scenes. (b) Violation of geometric constraints in a dynamic environment: (1) The tracked feature is too far from the epipolar line; (2) Back-projected rays from the tracked features do not meet; (3) The dynamic feature has an impact on the accuracy of the basic matrix estimation; (4) The reprojected feature is too far away from the observed feature

Figure 3(a) shows that the feature point transformation in the static background satisfies the epipolar geometric constraint. When the point x_1 changes to the point x_2 in the static scene, the epipolar constraint is:

$$x_2^T F x_1 = 0 \tag{1}$$

To filter out the feature points of motion more effectively, our solution process is as follows: first, we extract the feature points of the previous frame. Secondly, calculate the corresponding displacements of these points in the current frame by using the LK optical flow method of pyramid layering. Then, we estimate the fundamental matrix by the RANSAC algorithm through the previous frame's feature points and the corresponding points of the current frame. Next, use the fundamental matrix to calculate the epipolar line in the current frame. Finally, the matching point's distance to its corresponding epipolar line is calculated, and the motion state of the feature point is judged by comparing it with the preset threshold.

The feature points of the previous frame are projected into the current frame through the fundamental matrix. Let P_1 and P_2 denote the matching points in the previous frame and the current frame, respectively, and their homogeneous coordinate forms are:

$$P_1(u_1, v_1, 1), P_2(u_2, v_2, 1) \tag{2}$$

Among them, u and v respectively represent the position of the point in the image. According to the principle of epipolar geometry, we can get the epipolar line l_1 through the fundamental matrix F and the point P_1. The expression is:

$$I_1 = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = FP_1 = F \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} \tag{3}$$

X, Y, Z in the expression represent the line vectors of the epipolar line. Moreover, the distance from the matching point to its corresponding epipolar line is determined as follows:

$$D = \frac{\left|P_2^T F P_1\right|}{\sqrt{||X^2|| + ||Y^2||}} \tag{4}$$

D represents the distance from the matching point to the epipolar line. Our moving consistency module determines whether the point is a motion point by calculating the distance and comparing it with our previous preset threshold. Finally, if the matching point's distance to its corresponding epipolar line is less than the threshold, we consider the feature point is static. In contrast, if the distance is greater than the threshold, we consider the matching point to be moving.

3.4 Remove Outliers

In the Mask R-CNN network, we detect and extract potential moving targets in the image. Nevertheless, these potential moving targets are not certainly in a moving state, such as a car parked on the side of the road. If all the feature points in the potential moving target are eliminated indiscriminately, the SLAM system may lose track due to the lack of features. Therefore, it is particularly important to combine the moving consistency detection results with semantic segmentation to judge the movement state detection's potential moving targets.

Thanks to the semantic segmentation network, we can quickly obtain the complete contours of potential moving targets. We judge the target's movement state by the number of moving feature points in the potential moving targets' contour. Suppose the number of moving feature points is bigger than the threshold. We regard the target as a moving object. We will delete all the feature points in the target's contour, then use the remaining feature points for pose estimation. In this way, we can accurately eliminate the outliers that will affect the attitude estimation, thereby improving the system's accuracy.

3.5 Semantic Map

The maps used in the SLAM system are divided into point cloud map and octree map. The advantage of the point cloud map is that it can be efficiently generated directly from RGB-D images and does not require additional processing. However, the point cloud map is usually large and carry too much useless information, such as shadows and wrinkles. Simultaneously, the point cloud map cannot handle dynamic objects because the point cloud map can only add points during the construction process. The octree map is more flexible and updatable than the point cloud map. Furthermore, the octree map can be stored more efficiently and is easy to navigate.

We maintain the octree map through the octree map thread in the system. This thread will combine the keyframes obtained in the tracking thread with the segmentation results obtained in the semantic segmentation thread. We use the transformation matrix and depth image in the keyframe to create a local point cloud map. The local point cloud map is convenient for the system to perform local BA operations. And we convert and store the local point cloud map in the global octree map. The semantic information is merged into the octree map by binding the octree map's voxels to a specific color. We assign every semantic label to each different color. For example, red represents people, and blue represents cars, etc. In this way, the semantic information in the map can be updated efficiently.

In General, what is saved in the map should be the static background in the environment, so dynamic objects should not exist on the map. We can use semantic segmentation results to filter out dynamic objects. However, the accuracy of semantic segmentation is limited. In complex situations, for example, objects overlap each other, the semantic segmentation results may be incomplete or even wrong. We use a probability model to evaluate the possibility of a single voxel being occupied quantitatively to solve this problem. Let P denote the probability that a voxel n is occupied from time z_1 to z_t, and its expression is:

$$\mathrm{P}(n|z_{1:t}) = \left[1 + \frac{1-P(n|z_t)}{P(n|z_t)} \frac{1-P(n|z_{1:t-1})}{P(n|z_{1:t-1})} \frac{P(n)}{1-P(n)}\right]^{-1} \tag{5}$$

It can be seen that the value of this formula depends on the prior probability of P(n), $\mathrm{P}(n|z_{1:t-1})$, and $\mathrm{P}(n|z_t)$ at time z_t. By using the log-odds notation [22], it can be expressed as:

$$\mathrm{L}(n|z_{1:t}) = \mathrm{L}(n|z_{1:t-1}) + \mathrm{L}(n|z_t) \tag{6}$$

$$\mathrm{L}(n) = \log\left[\frac{P(n)}{1-P(n)}\right] \tag{7}$$

$L(n|z_{1:t})$ represents the log-odds score of voxel n from the start time z_1 to time z_t. When the voxel is repeatedly observed to be occupied, the voxel's log-odds score will increase. Otherwise, it will decrease. The occupied probability P of a voxel can be calculated by inverse logit transform. The state of the voxel is judged by comparing the probability P with our predefined threshold. When the probability P is greater than the threshold, we consider that the voxel is stably occupied. In other words, this voxel belongs to a static object. Through this method, we can handle the map construction problem in a dynamic environment well.

4 Experimental Results

In this experiment, we used the TUM RGB-D dataset as the test data and compared our solution with the original ORB-SLAM2. The TUM RGB-D data set [23] consists of 39 kinds of sequences recorded in different indoor scenes at full frame rate (30 Hz) using Microsoft Kinect sensors. The data set provides RGB-D images and real trajectories. In this dataset's walking sequence, there will be a large number of scenes of people moving. These moving people will significantly reduce the robustness and accuracy of the SLAM algorithm. This data set is highly dynamic. Therefore, it is challenging for the SLAM algorithm.

Figure 4(a) shows the evaluation of the absolute pose error (APE) between our measured value and the real value. This data evaluates the overall consistency of our measurement data with the real trajectory. Figure 4(b) Represents the real-time attitude error between our measured value and the real trajectory. The lower the color temperature of the track color, the closer our measured value is to the real value. The higher the color temperature of the track color, the greater the deviation between our measured value and the real value. Figure 5(a) shows the evaluation of the relative pose error (RPE) between our measured value and the real value. This data evaluates the translation and rotation drift between our scheme and the real trajectory. Figure 5(b) Represents the real-time translation drift error and rotation drift error between our measured value and the real trajectory. The lower the color temperature of the track color, the smaller the error.

Figure 6(a) shows the difference in our scheme, ORB-SLAM2, and the real trajectory. It reflects the system's integral accuracy. Figure 6(b) shows the deviation of our scheme and ORB-SLAM2 in the x, y, and z directions compared to the real trajectory. Figure 6 (c) shows the error between the two methods and the real trajectory attitude. Moreover, Tables 1, 2 and 3 show the comparison of rotation deviations and translation deviations in different schemes. It is clear that our solution is more robust and accurate in a dynamic environment than the original ORB-SLAM2.

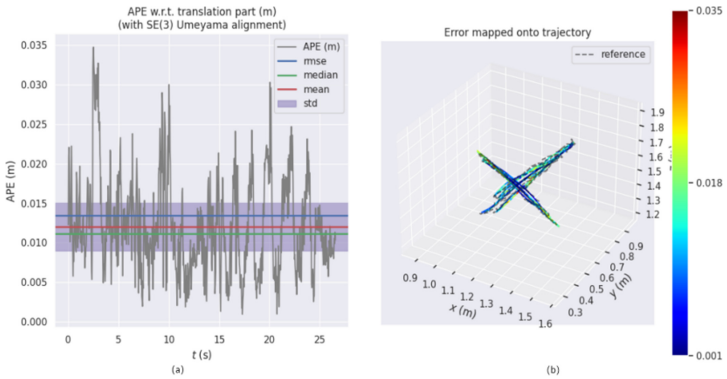

Fig. 4. (a) The absolute pose error (APE) between the measured value and the trajectory includes root mean square error (RMSE), median, and mean. (b) The deviation of the measurement data on the track position

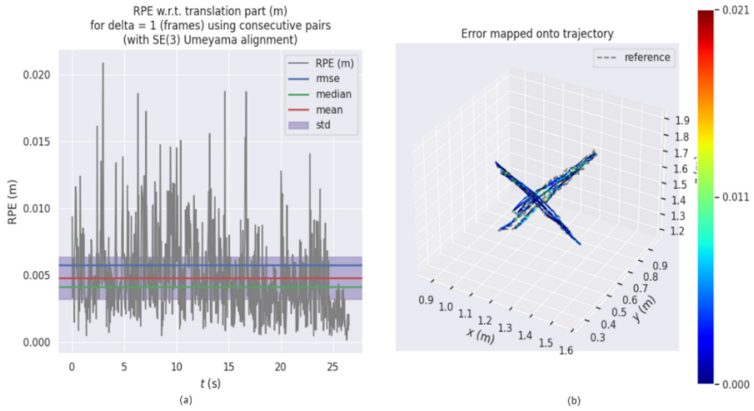

Fig. 5. (a) The relative pose error (RPE) between the measured value and the trajectory includes root mean square error (RMSE), median, and mean. (b) The deviation of the measurement data on the track position

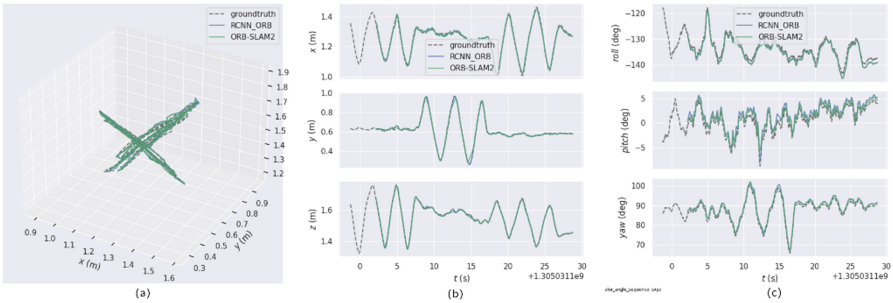

Fig. 6. (a) Our method and ORB-SLAM2 are compared with the real trajectory. (b) Compared with the real trajectory, the deviation of the two schemes in the x, y, and z directions. (c) The difference in attitude estimation between the two schemes compared to the real trajectory

Table 1. Results of metric rotational drift (RPE)

Sequences	ORB-SLAM2			Our Scheme		
	RMSE	Mean	Median	RMSE	Mean	Median
fr3_walking_xyz	7.7424	5.8754	4.5440	0.9234	0.5836	0.4197
fr3_walking_static	3.8754	1.5744	0.4571	0.2975	0.2415	0.2276
fr3_sitting_static	0.2887	0.2559	0.2495	0.2775	0.2417	0.2355

Table 2. Results of metric translational drift (RPE)

Sequences	ORB-SLAM2			Our Scheme		
	RMSE	Mean	Median	RMSE	Mean	Median
fr3_walking_xyz	0.1475	0.1254	0.1152	0.0013	0.0012	0.0012
fr3_walking_static	0.2167	0.0901	0.0154	0.0115	0.0091	0.0084
fr3_sitting_static	0.0095	0.0083	0.0074	0.0084	0.0078	0.0072

Table 3. Results of metric absolute trajectory error (ATE)

Sequences	ORB-SLAM2			Our Scheme		
	RMSE	Mean	Median	RMSE	Mean	Median
fr3_walking_xyz	0.7541	0.6484	0.5864	0.0247	0.0194	0.0172
fr3_walking_static	0.4052	0.3574	0.3028	0.0085	0.0075	0.0067
fr3_sitting_static	0.0087	0.0075	0.0062	0.0065	0.0054	0.0048

5 Conclusion

In this paper, we discussed some of the SLAM problems, firstly the accuracy problems in dynamic environments, secondly the limitations of geometric maps. We have proposed a scheme to solve these problems. We use the semantic segmentation technology of the deep learning network to capture potential dynamic objects in the image and use geometric constraints to interpret the potential moving objects' real state. SLAM's accuracy in a dynamic environment is improved by removing feature points from moving objects. Simultaneously, we extracted semantic information from the segmented image and combined it with traditional geometric maps to realize SLAM's perception of environmental information. After testing, our solution has significantly improved positioning accuracy and robustness compared to ORB-SALM2 in a dynamic environment. However, there are still some problems with our system. For example, static objects at the edge of images may be misunderstood as dynamic objects because they disappear from the next frame's edge. It may lead to a decrease in the system's accuracy, which is also the direction of our efforts in the future.

References

1. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. IEEE Trans. Robot. **33**, 1255–1262 (2017)
2. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: large-scale direct monocular SLAM. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8690, pp. 834–849. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10605-2_54
3. Cadena, C., Carlone, L., Carrillo, H.: Past, present, and future of simultaneous localization and mapping: toward the robust-perception age. IEEE Trans. Robot. **32**(6), 1309–1332 (2016)
4. Xia, L., Cui, J., Shen, R.: A survey of image semantics-based visual simultaneous localization and mapping: application-oriented solutions to autonomous navigation of mobile robots. Int. J. Adv. Robotic Syst. **17**(3), 172988142091918 (2020)
5. Viola: Robust real-time object detection. Int. J. Comput. Vis. **57**(2), 87 (2001)
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, San Diego, CA, USA, pp. 20–25 (2005)
7. Felzenszwalb, F., Girshick, R., McAllester, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Patt. Anal. Mach. Intell. **32**, 9 (2009)
8. Girshick, R.: From rigid templates to grammars: object detection with structured models. University of Chicago, Chicago (2012)
9. Lin, T.Y., Doll, P., Girshick, R.: Feature pyramid networks for object detection, pp. 936–944. CVPR, Honolulu, HI, USA (2017)
10. Sunderhauf, N., Pham, T.T., Latif, Y., Milford, M., Reid, I.: Meaningful maps with object-oriented semantic mapping. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2017)
11. Vasudevan, S., Gchter, S., Nguyen, V.: Cognitive maps for mobile robots—an object based approach. Robot. Auton. Syst. **55**, 359–371 (2017)
12. Pronobis, A., Jensfelt, P.: Large-scale semantic mapping and reasoning with heterogeneous modalities. In: IEEE International Conference on Robotics & Automation, pp. 3515–3522 (2012)
13. Mccormac, J., Handa, A., Davison, A., Leutenegger, S.: Semanticfusion: dense 3D semantic mapping with convolutional neural networks, pp. 4628–4635 (2016)

14. Whelan, T., Salas-Moreno, R.F., Glocker, B., Davison, A.J., Leutenegger, S.: Elasticfusion: real-time dense slam and light source estimation. Int. J. Robot. Res. **35**(14), 1697–1716 (2016)
15. Hhnel, D., Triebel, R., Burgard, W.: Map building with mobile robots in dynamic environments. IEEE Int. Conf. Robot. Autom. **2**, 1557–1563 (2003)
16. Alcantarilla, P., Yebes, J., Almazan, A.: On combining visual SLAM and dense scene flow to increase the robustness of localization and mapping in dynamic environments. In: IEEE International Conference, pp. 1290–1297 (2012)
17. Tan, W., Liu, H., Dong, Z., Zhang, G., Bao, H.: Robust monocular SLAM in dynamic environments. In: IEEE International Symposium on Mixed & Augmented Reality (2013)
18. Kundu, A., Krishna, K., Jawahar, C.: Realtime multibody visual slam with a smoothly moving monocular camera. In: ICCV, pp. 2080–2087 (2011)
19. Wang, C., Thorpe, C., Thrun, S., Hebert, M., Durrant-Whyte, H.: Simultaneous localization, mapping and moving object tracking. IJRR **26**(9), 889–916 (2007)
20. Kaiming, H., Georgia, G., Piotr, D., Ross, G.: Mask R-CNN, pp. 1–1. IEEE Trans. Pattern Anal. Mach. Intell., PP (2017)
21. Saputra, M.R.U., Markham, A., Trigoni, N.: Visual slam and structure from motion in dynamic environments: a survey. ACM Comput. Surv. **51**(2), 1–36 (2018)
22. 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)
23. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. Intelligent Robots and Systems (IROS) (2012)