2022 International Conference on Computing, Communication, Perception and Quantum Technology (CCPQT) | 978-1-6654-7020-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/CCPQT56151.2022.00016

Research on Vision-based Semantic SLAM towards Indoor Dynamic Environment

1st Chun Yang
University of Science and Technology Beijing
Beijing, China
13283213637@163.com

2nd Ting Lyu*
University of Science and Technology Beijing
Beijing, China
lvting_z@163.com

***Abstract*—Most of the maps constructed based on traditional visual SLAM technology are sparse maps, which only contain geometric information and do not contain semantic information, which limits the robot to complete the tasks of understanding. In this paper, we propose a vision-based semantic SLAM method. The visual odometry is optimized by using semantic information to remove the influence of dynamic objects in the scene. Based on the proposed method, we can finally construct a semantic map. Experiments show that, our system effectively improves the positioning and mapping accuracy.**

Keywords- Visual SLAM; Semantic map; Dynamic feature point detection

I. INTRODUCTION

In the research of robot technology, simultaneous localization and mapping (SLAM occupies an important position**Error! Reference source not found.**, which is the basis for the robot to complete control, movement and other tasks. Specifically, it refers to the technology that the main body equipped with specific sensors establishes an environmental model in the process of movement without environmental prior information. Compared with expensive laser sensors, vision sensors use cameras to obtain external information, which can obtain rich environmental information at the same time of low price. However, most of the traditional maps built based on visual SLAM(VSLAM) technology are metric maps used to represent the location relationship in the map. Combining VSLAM with deep learning to build semantic maps can help robots understand complex instructions at the semantic level and provide information support for a series of tasks such as obstacle avoidance and path planning.

At the same time, the current research on VSLAM system often ignores the dynamic interference factors in the application scene. If the application scene is static, it can often build a high-precision map. However, in real life, there are often many external interference factors in the indoor scene. Due to the existence of these interference, it will be difficult to build a globally consistent map.

Therefore, combining the mature VSLAM framework with deep learning to build a semantic map containing object semantic attributes is of great significance for robot self-understanding. At the same time, to make VSLAM meet more indoor scenes, it is also of great significance to study how to remove the impact of dynamic objects on the accuracy of map construction.

II. RELATED WORK

A. Visual SLAM

In 2007, Davison et al. proposed the first monocular VSLAM system, named MonoSLAM**Error! Reference source not found.**. However, the calculation amount of this method will increase with the increase of the environmental range. Klein et al. proposed PTAM**Error! Reference source not found.** in the same year. PTAM proposed a multi-threading method for the first time. At the same time, PTAM combines the bundle adjustment (BA)**Error! Reference source not found.** with real-time VSLAM for the first time. The BA-based method can process a large number of feature points and build maps based on keyframes. Since PTAM was proposed, most VSLAM systems have adopted multi-threading and back-end nonlinear optimization methods. The ORB-SLAM was proposed by Mur-Artal et al. **Error! Reference source not found.** in 2015, which is a complete and mature SLAM system that can work on standard CPUs in various environments. The system can use ORB features to estimate camera motion. Mur-Artal et al. proposed ORB-SLAM2**Error! Reference source not found.** in 2017. This system supports multiple types of cameras and adds a global optimization thread on the basis of the original three threads, which has high positioning accuracy.

B. Semantic SLAM

In 2015, Semantic Fusion**Error! Reference source not found.** is proposed by McCormac et al. In 2016, Pham et al.**Error! Reference source not found.** used ORB-SLAM2 to estimate the pose of the camera [8] to detect the object in the keyframe and obtain the category and probability of objects, which applied unsupervised 3D segmentation method to obtain the corresponding point cloud sequence of each object, but there will be many missed detection results. In 2018, Bescos et al. constructed DynaSLAM**Error! Reference source not found.**, using Mask R-CNN**Error! Reference source not found.** to detect dynamic objects, complete the removal of dynamic objects, and remove the background of dynamic objects through previous image frame repair. In 2018, the DS-SLAM**Error! Reference source not found.** is proposed by Yu et al., which is a pixel level SLAM system for dynamic environment combining SegNet**Error! Reference source not found.** with a moving consistency check. In 2020, Cui et al.**Error! Reference source not found.** realized the target detection of objects in the environment. Aiming at the disadvantage of poor stability of ORB-SLAM2 under weak light conditions, they used thermal infrared image and depth image.

978-1-6654-7020-9/22/$31.00 ©2022 IEEE
DOI 10.1109/CCPQT56151.2022.00016

III. SYSTEM DESCRIPTION

A. System Framework

Aiming at the problems existing in the above system, we design a improved and optimized SLAM system based on ORB-SLAM2. As shown in Figure 1, two new threads are added on the basis of the original four threads: the semantic segmentation thread, the semantic mapping thread. The tracking thread have been optimized and improved.

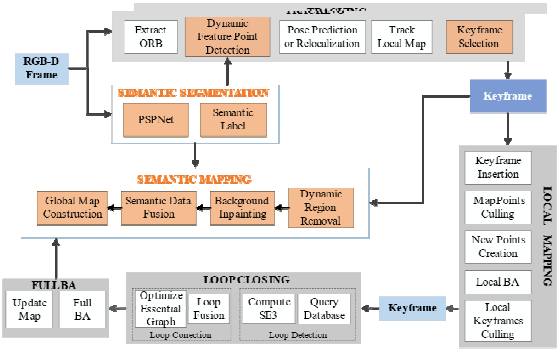


Figure 1. System framework

B. Semantic Segmentation

In view of the changeable scenes in the indoor environment, and the objects in the scene are of various types and sizes, this paper selects PSPNet**Error! Reference source not found.** as the semantic segmentation network used by the system. The pyramid module proposed by the PSPNet can obtain global information, so as to combine local and global context information, and associate semantic labels through the relationship between categories, so as to make the prediction results more real and reliable.

This paper uses the dataset PASCAL VOC2012 used in the official manual of PSPNet as the training sample of the network. Pascal VOC2012 includes 21 categories including background, the common indoor objects include bottle objects, chairs, dining tables, sofas, potted plants, televisions, screens , monitors, and people, cats and dogs can walk indoors.

C. Tracking Thread Optimization

The optimization of the original tracking thread mainly includes two parts:

- After feature matching, a dynamic feature point removal module is added to avoid drift or even loss of pose estimation.
- The keyframe selection method is designed. Under the current loose selection conditions, add constraints to reduce the generation of map redundant information, avoid the loss of tracking, and improve the mapping accuracy.

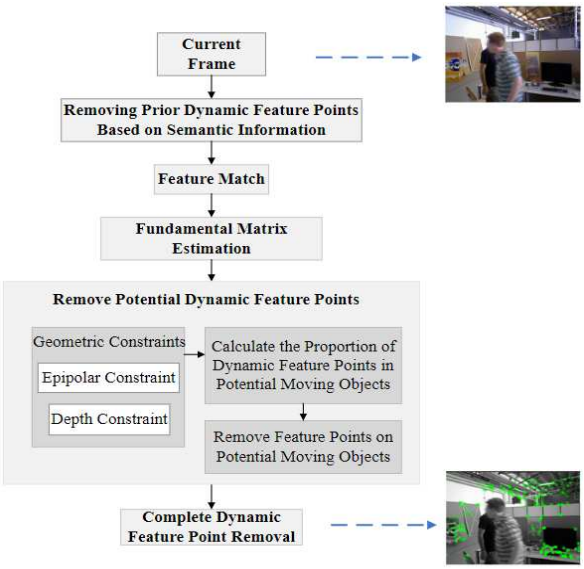


Figure 2. Flow chart of dynamic feature point removal

1) Dynamic Feature Point Detection: A dynamic feature point removal method combining semantic information and multiple-view geometric constraints is proposed, where the flow chart of alorithm is shown in Figure 2.

a) Geometric Constraints: Epipolar geometry[16] is a two-view geometric relationship that constrains the projection of spatial points on different imaging planes. Ideally, the matching feature point pair of static points satisfies the fundamental matrix F . However, when the spatial point p is dynamic (as shown in Fig. 3), there will be an error in the solved F , which will lead to a certain distance d between the correctly matched feature points and the ideally matched feature points (epipolar).

Let the polar equation be: $Ax+By+C=0$, then the distance d from p_2 to the polar can be calculated by the (1):

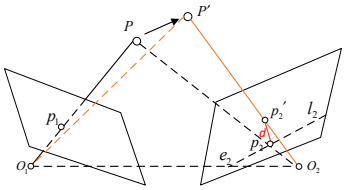


Figure 3. Epipolar constraint error of dynamic points

$$d=\frac{\left|Au_2^{p_2}+Bv_2^{p_2}+C\right|}{\sqrt{\square A\square^2+\square B\square^2}}=\frac{\left|p_2^T F p_1\right|}{\sqrt{\square A\square^2+\square B\square^2}} \quad (1)$$

However, for dynamic points in parallel with the camera motion (that is, when the camera motion and the dynamic object are relatively stationary), their projection will also fall on the epipolar, as shown in Fig.4. Therefore,

this paper adds a depth constraint to constrain such dynamic feature points.

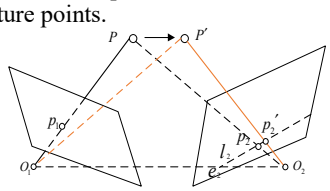


Figure 4. Epipolar Constraint Degenerate Case

Set the feature point $p_1(u_1, v_1, 1)$ in the image I_1, and its corresponding spatial point is $P_w(X_w, Y_w, Z_w)$. The coordinate $P(X, Y, Z)$ of the point in the camera coordinate system of the image I_2 can be expressed as $P = RP_w + T$:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{2}$$

The value of theoretical projection point depth Z can be obtained as (3):

$$Z = R_{31}X_w + R_{32}Y_w + R_{33}Z_w + t_z \tag{3}$$

The matching point p_2 of p_1 in image I_2 can be obtained through feature matching, and its actual depth value is Z' through RGB-D camera. Therefore, the constraint equation of depth error can be obtained:

$$d = Z - Z' \tag{4}$$

Taking the epipolar error and depth error together as the condition to judge whether any feature point is a dynamic feature point, its dynamic value can be defined as:

$$D = |d_1| + |d_2| \tag{5}$$

Where d_1 is the epipolar error value and d_2 is the depth error value. In the experiment, the threshold value θ is set to 0.1.

b) Method Detail: Suppose a set of feature points $Q = \{P_i \in \mathrm{R}^2\}_{i=1}^{n}$ in the input image, all feature points are marked as static feature points during initialization. After the dynamic feature point detection algorithm, the dynamic feature points and static feature points can be separated to form two sets, $Q_d = \{q_1, q_2, \cdots, q_m\}$ and $Q_s = \{q_{m+1}, q_{m+2}, \cdots, q_n\}$, $Q_d \cap Q_s = \varnothing$.

- The feature points can be divided into three categories: dynamic category $D = \{d_1, d_2 \cdots\}$, potential dynamic category $PD = \{pd_1, pd_2 \cdots\}$, static category $S = \{s_1, s_2 \cdots\}$. The dynamic category includes objects that can move by themselves, such as people and animals; The potential dynamic category includes movable objects, such as chairs and computers; Static category includes objects that cannot move, such as walls, floors, etc.
- Mark the feature points in the dynamic category and add them to the dynamic feature point set Q_d. The remaining feature points are added to the set of static feature points Q_s, and the fundamental matrix F is calculated by using the feature points in Q_s, thereby ignoring the influence of a priori dynamic feature points on feature matching.
- The camera pose T can be obtained by using the F, and use (5) to calculate the dynamic value of the feature points in PD. When the dynamic value exceeds the set threshold, the current feature point p_i is considered to be dynamic. Then calculate the number of dynamic feature points in the category pd_i where p_i is located.

2) Keyframe Selection: We propose a keyframe selection method based on inter-frame motion and static co-view index. First, use the camera matrix T to calculate the translation distance t and rotation angle θ of the movement, and judge whether the camera has performed effective movement according to (6):

$$\begin{cases} \|\theta\| > \theta_{th} \\ \|t\| > t_{th} \end{cases} \tag{6}$$

where, θ_{th} and t_{th} are the set thresholds.

But it is not enough to only determine that the camera has made valid motion, because the pose also changes when the camera moves back and forth, but there is no need to add new keyframes.

Therefore, after determining that the camera has made effective motion, considering the possible dynamic objects in the scene, calculate the common view index of the static feature points of the current frame and the previous keyframe to judge whether it is lower than the set threshold, at the same time, determine whether the number of new feature points exceeds 15. The formula for calculating the static common view index between two frames is defined as (7):

$$\mu = \frac{N_1 \cap N_2}{N_1} \tag{7}$$

Where, N_1 and N_2 is the number of static observation points of the two images, $N_1 \cap N_2$ is the number of static observation points that can be observed in both images.

D. Semantic Mapping

1) Dynamic Region Removal and Background Inpainting: Using the dynamic region detection method of frame mentioned before, the dynamic region in the keyframe of image construction is removed. As shown in Fig.5, so the map can be constructed without using the map points on the dynamic region.

(a) RGB original keyframes

(b) Keyframes of dynamic region removal

Figure 5. Dynamic region removal

After removing the dynamic region of the keyframe, this paper uses the background inpainting method mentioned in reference **Error! Reference source not found.** by using the previous keyframe. If it is an area that has not appeared in the previous keyframe.

2) Semantic Data Fusion: Through the tracking of the camera movement by the SLAM system, the camera pose of each keyframe can be obtained, and then the semantic category probability value of the map point can be assigned according to the semantic information obtained by the semantic segmentation thread. And finally the attribute value $\varphi_i = \{P_i, I_i, D_i, K_i\}$ of each map point P can be obtained, where P_i is the spatial coordinates , I_i is the RGB value, D_i is the depth information, K_i is the semantic category.

In this paper, the Bayesian-based method proposed in **Error! Reference source not found.** is used to update the semantic information of map points. In order to avoid too much category information stored in map points and waste of resources when there are too many object categories trained with the semantic segmentation network, an improved method is proposed: firstly, for each pixel Q_i in the keyframe K_t , only the first two categories l_k and l_m with the highest probability value of semantic category are recorded, and the semantic category set of the point is defined as $L_i = \{l_k, l_m\}$. Then, the set category is updated by iteration. When there are common view map points in two keyframes, if the semantic category set of the point under the current frame is consistent with the category in the existing semantic category set, the probability of semantic category in the semantic category set is updated by Bayesian formula. If the semantic category set of the point under the current frame is inconsistent with the category in the existing semantic category set, the new semantic category l_n is added to the semantic category set of the point. The newly added semantic category probability is also updated by Bayesian formula. Finally, the construction of globally consistent semantic point cloud map is completed. As shown in Figure 6, the high dynamic sequence walking_xyz of TUM are used to construct the point cloud map based on ORB-SLAM2, cloud map and semantic map with dynamic objects removed constructed by using the system in this paper.

(a)point cloud map (b)point cloud map with-out dynamic objects (c) semantic map

Figure 6. Map of fr3/walking_xyz

IV. Experiment Results

A. Datasets and Evaluation Metrics

The TUM dataset**Error! Reference source not found.** is an RGB-D dataset recorded by Microsoft Kinect sensors at 30fps in different indoor environments, while providing the ground truth of camera trajectories. We use low-dynamic sequences and high-dynamic sequences in the fr3 sequences of TUM for experiments, in which dynamic objects in low-dynamic sequences have been sitting in their seats and communicate with small motion amplitudes, while dynamic objects in high-dynamic sequences walk around desks with motion amplitudes very large.

B. System Evaluation

The ATE plots converts 3D camera space trajectories and ground truth trajectories into 2D plane trajectories and draws them in the same figure, evaluating the systematic error by observing the difference between the two trajectories. As shown in Fig. 7 - 9，in sitting_xyz, the semantic SLAM we proposed has little improvement in accuracy compared with ORB-SLAM2. This is because the dynamic objects do not move significantly and ORB-SLAM2 tracks and maps the

TABLE I. Comparison of Absolute Trajectory Error (ATE)

Sequences	ORB-SLAM2				DynaSLAM				DS-SLAM				Our System			
	RMSE (m)	*Mean (m)*	*Median (m)*	*S.D. (m)*	*RMSE (m)*	*Mean (m)*	*Median (m)*	*S.D. (m)*	*RMSE (m)*	*Mean (m)*	*Median (m)*	*S.D. (m)*	*RMSE (m)*	*Mean (m)*	*Median (m)*	*S.D. (m)*
stting_static	0.0087	0.0076	0.0068	0.0041	0.0091	0.0080	0.0075	0.0043	0.0065	0.0055	0.0049	0.0033	**0.0056**	**0.0049**	**0.0044**	**0.0027**
walking_static	0.3927	0.3597	0.3015	0.1577	0.1510	0.1360	0.1123	0.0656	**0.0081**	0.0073	0.0067	**0.0036**	0.0089	**0.0073**	**0.0044**	0.0044
walking_halfsphere	0.4620	0.4281	0.4234	0.1736	0.0332	0.0282	0.0244	0.0176	0.0303	0.0258	0.0222	0.0159	**0.0256**	**0.0224**	**0.0199**	**0.0124**
walking_xyz	0.6428	0.5315	0.4434	0.3615	0.0179	0.0150	0.0126	0.0097	0.0247	0.0186	0.0151	0.0161	**0.0154**	**0.0134**	**0.0119**	**0.0075**

TABLE II. COMPARISON OF TRANSLATIONAL RELATIVE POSE ERROR (RPE)

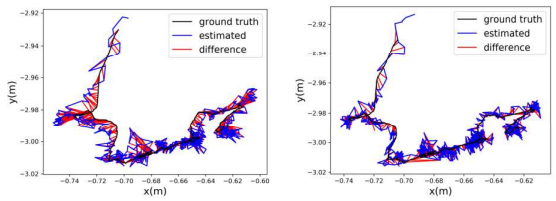
Sequences	ORB-SLAM2				DynaSLAM				DS-SLAM				Our System			
	RMSE (m)	*Mean (m)*	*Median (m)*	*S.D. (m)*	*RMSE (m)*	*Mean (m)*	*Median (m)*	*S.D. (m)*	*RMSE (m)*	*Mean (m)*	*Median (m)*	*S.D. (m)*	*RMSE (m)*	*Mean (m)*	*Median (m)*	*S.D. (m)*
stting_static	0.00934	0.0907	0.0074	0.0044	0.01150	0.0100	0.0089	0.0056	0.0078	0.0068	0.0061	0.0038	**0.0071**	**0.0064**	**0.0060**	**0.0031**
walking_static	0.21112	0.0907	0.0134	0.1907	0.0834	0.0429	0.0119	0.0714	**0.0102**	**0.0091**	**0.0082**	**0**.0048	0.0114	0.0100	0.0089	0.0056
walking_halfsphere	0.3728	0.2124	0.0519	0.3063	0.0355	0.0310	0.0279	0.0173	0.0297	0.0256	0.0226	0.0152	**0.0243**	**0.0215**	0.0198	**0.0112**
walking_xyz	0.4532	0.3135	0.2419	0.3272	0.0222	0.0191	0.0166	0.0114	0.0333	0.0238	0.0181	0.0229	**0.0187**	**0.0164**	**0.0145**	**0**.0092

TABLE III. COMPARISON OF ROTATIONAL RELATIVE POSE ERROR (RPE)

Sequences	ORB-SLAM2				DynaSLAM				DS-SLAM				Our System			
	RMSE (m)	*Mean (m)*	*Median (m)*	*S.D. (m)*	*RMSE (m)*	*Mean (m)*	*Median (m)*	*S.D. (m)*	*RMSE (m)*	*Mean (m)*	*Median (m)*	*S.D. (m)*	*RMSE (m)*	*Mean (m)*	*Median (m)*	*S.D. (m)*
stting_static	0.2850	0.2581	0.2474	0.1208	0.4746	0.4043	0.3516	0.2485	0.2735	0.2450	0.2351	0.1215	**0.2645**	**0.2369**	**0.2219**	**0.1176**
walking_static	3.7846	1.6666	0.3387	3.3979	1.4314	0.7411	0.2727	1.2246	0.2690	0.2416	0.2259	0.1182	**0.2546**	**0.2043**	**0.2216**	**0.1087**
walking_halfsphere	7.7882	4.4612	1.3567	6.3839	0.8942	0.7707	0.6751	0.4536	0.8142	0.7033	0.6217	0.4101	**0.7231**	**0.6379**	**0.5680**	**0.3407**
walking_xyz	8.5088	5.7949	3.7396	6.2305	0.6578	0.5306	0.4464	0.3887	0.8266	0.5836	0.4192	0.5826	**0.6003**	**0.4675**	**0.3861**	**0.3765**

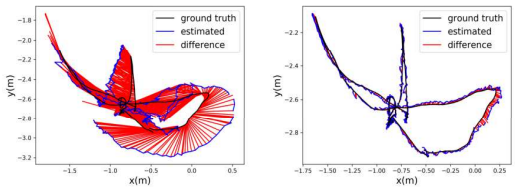
dynamic object as a static object. But because the dynamic objects in low dynamic scene also have small movements, the ATE and RPE of our system are smaller than ORB-SLAM2 the evaluation is more accurate. In walking_halfsphere , the system accuracy of ORB-SLAM2 is greatly reduced, the estimated camera trajectory is very different from the actual trajectory, and the relative pose error is up to 1.4m. And the semantic SLAM proposed effectively reduces the system error. The estimated trajectory is very close to the ground truth trajectory, and the relative pose error is controlled within 0.08m, which effectively improves the system accuracy.

In order to make a better qualitative analysis of our system, calculate the RMSE, median, mean and S.D. of ATE and RPE, and compared with ORB-SLAM2, DynaSLAM and DS-SLAM and recorded in Table I- III.



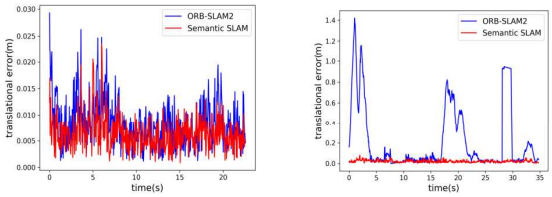
(a) ORB-SLAM2 (b) semantic SLAM

Figure 7. ATE of sitting_xyz



(a)ORB-SLAM2 (b)semantic SLAM

Figure 8. ATE of walking_halfsphere



(a) sitting_xyz (b) walking_halfsphere

Figure 9. Comparison of RPE

As shown in Table I- III, in the low dynamic scene, the performance of our system is similar to that of the other three systems, but because our system can detect static dynamic objects, the ATE and RPE values are lower. Under the high dynamic sequence, it is obvious that the systematic error of ORB-SLAM2 is large. DynaSLAM, DS-SLAM and our system can effectively reduce the systematic error. Since we propose a geometric constraint method combining multiple constraints, the system error is smaller. In terms of the RMSE of the ATE of the sequence walking_static, walking_halfsphere, walking_xyz, it is 61.5%, 94.5%, and 97.6% lower than ORB- SLAM2, and the other three indicators are reduced to a similar degree to the RMSE.

V. CONCLUSION

Aiming at the problems existing in traditional VSLAM, we combine traditional VSLAM with deep learning. Using this semantic information, the tracking thread is optimized in two aspects: first, a dynamic feature point detection module is added, and the semantic information is combined with the geometric constraint method of multiple constraints to remove the dynamic feature points;

the second is to design the keyframe selection method avoids tracking loss by calculating the common view index of static feature points. Using the selected keyframes, a pixel level point cloud semantic map is constructed. The map gives each pixel with semantic information and removes dynamic regions. Experiments are conducted with the TUM dataset and compared with some state-of-the-art systems, and the experimental results show that our system has better localization and mapping accuracy in dynamic environments.

ACKNOWLEDGMENT

This work is supported by the key research and development plan project of Guangdong Province (No.2020B0909020001).

REFERENCES

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," in IEEE Robotics & Automation Magazine, vol. 13, no. 2, pp. 99-110, June 2006, doi: 10.1109/MRA.2006.1638022.

[2] R. A. Newcombe, S. J. Lovegrove and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," 2011 International Conference on Computer Vision, 2011, pp. 2320-2327, doi: 10.1109/ICCV.2011.6126513.

[3] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2007, pp. 225-234, doi: 10.1109/ISMAR.2007.4538852.

[4] B Triggs, P F McLauchlan, R I Hartley, "Bundle adjustment—a modern synthesis." International workshop on vision algorithms, 1999, pp. 298-372.

[5] R. Mur-Artal, J. M. M. Montiel and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," in IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163, Oct. 2015, doi: 10.1109/TRO.2015.2463671.

[6] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," in IEEE Transactions on Robotics, vol. 33, no. 5, pp. 1255-1262, Oct. 2017, doi: 10.1109/TRO.2017.2705103.

[7] J. McCormac, A. Handa, A. Davison and S. Leutenegger, "SemanticFusion: Dense 3D semantic mapping with convolutional neural networks," 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 4628-4635, doi: 10.1109/ICRA.2017.7989538.

[8] N. Sünderhauf, T. T. Pham, Y. Latif, M. Milford and I. Reid, "Meaningful maps with object-oriented semantic mapping," 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 5079-5085, doi: 10.1109/IROS.2017.8206392.

[9] W. Liu, D. Anguelov, D. Erhan, S. Christian, S. Reed, C.-Y. Fu, et al., "SSD: single shot multibox detector," European conference on computer vision , 2016, pp. 21-37.

[10] B. Bescos, J. M. Fácil, J. Civera and J. Neira, "DynaSLAM: Tracking, Mapping, and Inpainting in Dynamic Scenes," in IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 4076-4083, Oct. 2018, doi: 10.1109/LRA.2018.2860039.

[11] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980-2988, doi: 10.1109/ICCV.2017.322.

[12] C. Yu, Z Liu, X J Liu, "DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 1168-1174, doi: 10.1109/IROS.2018.8593691.

[13] V. Badrinarayanan, A. Kendall and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2481-2495, 1 Dec. 2017, doi: 10.1109/TPAMI.2016.2644615.

[14] X. Cui, C. Lu and J. Wang, "3D Semantic Map Construction Using Improved ORB-SLAM2 for Mobile Robot in Edge Computing Environment," in IEEE Access, vol. 8, pp. 67179-67191, 2020, doi: 10.1109/ACCESS.2020.2983488.

[15] H. Zhao, J. Shi, X. Qi, X. Wang and J. Jia, "Pyramid Scene Parsing Network," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6230-6239, doi: 10.1109/CVPR.2017.660.

[16] R Hartley, A Zisserman. Multiple view geometry in computer vision. Cambridge university press, 2003.

[17] L. Ma, J. Stückler, C. Kerl and D. Cremers, "Multi-view deep learning for consistent semantic mapping with RGB-D cameras," 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 598-605, doi: 10.1109/IROS.2017.8202213.

[18] L. Ma, J. Stückler, C. Kerl and D. Cremers, "Multi-view deep learning for consistent semantic mapping with RGB-D cameras," 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 598-605, doi: 10.1109/IROS.2017.8202213.