PROB-SLAM: Real-time Visual SLAM Based on Probabilistic Graph Optimization

Xianwei Meng*
School of Mechanical Engineering, Shanghai Jiao Tong University, China
mengxianwei991203@sjtu.edu.cn

Bonian Li
584119429@qq.com

1 INTRODUCTION

Over the past several decades, with the development of science and technology, the use of computers for simultaneous localization and mapping (SLAM) has become an indispensable part of many automation fields. It enables the robot to have the ability to locate and obtain the location information of the environment in a new environment, so it is a crucial technology for autonomous navigation of mobile robots [1] and is widely used in autonomous driving [2], drones and augmented reality (AR) [3] and other fields.

In the past several decades, the SLAM has been a widely used research direction, so many excellent visual SLAM systems have emerged, such as ORBSLAM [4], ORB-SLAM2 [5], LSD-SLAM [6], and SVO [7]. When the environment is static, or there are few dynamic elements, these excellent systems can achieve satisfactory performance. However, unstable environmental factors are ubiquitous, and sometimes the movement of people in indoor positioning will affect the positioning results. Due to static world assumptions and lighting or other influences, the above standard SLAM systems are significantly less accurate and less robust in such highly dynamic environments.

An important branch of SLAM, dynamic SLAM algorithms, can be a hot research area today. This technology can already be used in many real-world applications, such as mobile robotics. For traditional dynamic SLAM algorithms, Cadena et al. provided a more comprehensive overview of the past, present, and future of SLAM [8]. However, this type of dynamic SLAM has no prior semantic information. In the absence of semantic information as a priori, using reliable constraints to find the right feature matching relationship is the primary method to deal with dynamic SLAM problems. Li et al. proposed a static weighting method for keyframe edge points and integrated it into the IAIQP method to reduce tracking error [9] Jin et al. proposed a dense visual odometry calculation method based on a background model, estimated from the depth scene Nonparametric Background Model [10]. Dai et al. distinguished dynamic and static map points based on feature correlation [11]. Flow fusion used optical flow residuals to highlight dynamic regions in RGB-D point clouds [12]. Since deep learning networks do not need to provide semantic priors, the above methods are generally fast in handling dynamic factors but lack accuracy. In response to this problem, VSO [13] proposed a novel method. If the probability of each object in the picture is obtained based on the semantic segmentation of a particular frame, that is, the probability of using the object is inversely proportional to the distance from the center of the object. The concept of distance can be used to create the concept of normalization, which reduces the background part that is far away in semantic segmentation, thereby improving the accuracy of
pose estimation, but because the error of semantic segmentation itself is more prominent, and semantic segmentation often exist gaps between different instances, resulting in lower overall accuracy, so this method is not very helpful for practical results.

Specifically, in deep learning methods, the most common techniques in the deep learning section [14] are usually adopted to generate the necessary semantic information, such as object detection and semantic segmentation techniques. Li et al. used the well-known semantic segmentation network SegNet to segment images, which were further processed to distinguish dynamic objects [15]. Zhang et al. adopted YOLO [16] to identify objects in the environment and built a semantic map to filter dynamic feature points to improve the system’s accuracy. Liu et al. adopted different semantic segmentation methods to detect dynamic objects and remove outliers [17]. Deep learning methods are helpful when dynamic objects in the environment are known in advance. However, deep learning methods rely heavily on the quality of the network [18]. A network with a simple architecture may be unable to effectively identify dynamic objects in some cases, while a complex network architecture may slow down the system.

A subset of SLAM also applies CNNs to segment previously dynamic objects in images and re-examine them using multi-view geometric models. Zhao et al. [19] adopted a Mask-RCNN network to discover the contours of latent dynamic objects. They then checked the consistency of optical flow to detect the actual state of dynamic regions. Yang et al. [20] first used an object detection model to detect predefined dynamic objects and then adopted multi-view constraints to confirm dynamic pixel depth images. However, semantic segmentation, such as SegNet and Mask-RCNN, is time-consuming, and multi-view methods require multiple frames of images. This makes the SLAM system face practical and stability challenges in practical applications.

In detail, SLAM generally has two basic requirements: robustness of tracking and real-time performance. Therefore, detecting dynamic objects that populate the scene and preventing tracking algorithms from using data associations related to these dynamic objects in real-time is a challenge that enables SLAM to be deployed in the real world. With the development of deep learning technology, many studies have introduced semantic information as a constraint to solve dynamic problems in SLAM. By means of object recognition and semantic segmentation, constraints are added to the feature point information distributed on objects. SLAM performance tends to be significantly improved. First, Ryan R et al. proposed the concept of probabilistic semantics. They added geometry, semantics, and sensors into the same framework for optimization and solved it with the EM algorithm, realizing a SLAM system with higher positioning accuracy. The sensor and the lens are used to observe the correspondence between the road signs and the road signs, and the weight factor represents it. The probability model calculates the object’s center, and then the second image projection is performed to approach the center of the detection frame. Konstantinos-Nektarios Lianos et al. [13] proposed a VSO algorithm. They proposed to use semantic segmentation to obtain the probability of each object in the picture. However, the boundary of semantic segmentation often lacked accuracy, and the real-time performance was also poor. These defects are more significant when there are dynamic objects in the scene, so the practice of introducing semantic information into SLAM cannot be used in most scenes with dynamic objects.

In response to the above problems, we tightly organize deep learning and static SLAM. At the same time, the two-dimensional Gaussian distribution is used to process the probability distribution map, which effectively reduces the negative impact of inaccurate semantic segmentation borders, realizes the filtering of feature points distributed on dynamic objects, and retains sufficient information on static objects to the greatest extent. To adjust the weight of feature points distributed on dynamic objects in optimization so that the dynamic SLAM system is more stable and less affected by moving objects - we propose PROB-SLAM, which is based on a High-performance, visual SLAM system for object detection and probability distribution maps in dynamic indoor environments. Based on ORB-SLAM2 [5], this Prob-SLAM system uses YOLO [16] to identify dynamic objects based on predefined knowledge, obtains semantic information through YOLOv5, and distinguishes high-confidence regions from the boundary range obtained by semantic segmentation. Gaussian distribution is introduced into the high-confidence region, a probability distribution map is finally constructed. SLAM is optimized from a new dimension, and the attribute of confidence of critical points is given according to the probability distribution map under various constraints. Next, this paper calculates the feature point information required for pose estimation. The confidence of the critical points is used as a weight to participate in the camera pose optimization. In this way, the goal of reducing the importance of dynamic objects in SLAM and retaining the information of static objects to the greatest extent is achieved, which makes good use of semantic information and reduces the negative impact of inaccurate semantic segmentation. To verify the effectiveness of PROB-SLAM, we conduct dynamic monocular SLAM experiments on the TUM dataset. Our method achieves higher-than-original localization accuracy in both low-dynamic and high-dynamic scenes on the same dataset compared with the original ORB-SLAM. The main contributions of this paper are as follows:

- A probabilistic algorithm based on the foreground (within the bounding box) and background (outside the bounding box) information is attached to the key points. Aiming at the error problem caused by dynamic objects, a PROB algorithm based on semantic segmentation is proposed to weaken the influence of dynamic objects and preserve the influence of static objects.
- Based on the YOLOv5 target detection network and predefined knowledge, this paper distinguishes between dynamic objects and static objects, distinguishes the picture frame’s foreground and background points, and perform a probability algorithm based on two-dimensional Gaussian distribution for the foreground points to minimize the inaccurate boundary of semantic segmentation. With the coming influence, the final construction gets the probability distribution map.
- The probability distribution map constructed according to the target motion attribute weakens the optimization weight of the dynamic object, provides it to the subsequent optimization algorithm and processes it in the subsequent newly added PROB thread, and uses the probability distribution
we directly regard dynamic objects as part of the background. We focus on based on predefined knowledge and this paper
constructed a set of static objects to reduce the probability of feature mismatch. Therefore, it is more reliable for localization and
tained in these stationary objects can maintain better stability after
shapes contain more features, such as sharper edges or corners.
In the SLAM process, Xiang Gao et al. [21] proposed that, it is
usually necessary to infer the pose and positioning according to
the noise due to various interference and misrecognition noise.
According to the confidence distribution map, we assign, Attribute
prob$_{i,j}$, confidence to each pixel of each frame of the picture. When
the feature point extraction is completed, and we can optimize the
back end according to the confidence attribute of the feature point
in Figure 2. Since SLAM uses G2O optimization to solve the Bundle
Adjustment problem, we multiply the probability attribute into the
covariance matrix corresponding to each error term of the least
squares problem constructed by the Bundle Adjustment problem
to introduce the probability attribute to each error term. Weights
use yolov5 according to the set of static objects. Performs scene
semantic segmentation.

2.2 Construction of confidence distribution map
We construct a confidence distribution map based on the position of the center point of the static object obtained by scene semantic
segmentation. We try two distributions, normal and Gaussian
distributions, and compare the robustness and accuracy of the two
distributions to our method. This paper selects Gaussian as the
distribution. Introducing a Gaussian distribution into the bounding
box can effectively alleviate the negative impact of inaccurate semantic segmentation. The two-dimensional Gaussian distribution
can minimize the impact of non-target static objects within the
bounding box on subsequent pose estimation. We take the center of the stationary object obtained from the scene semantic segmentation as the center of the two-dimensional Gaussian distribution, assign the center of the two-dimensional Gaussian distribution with
a confidence of 0.99, and assign the edge of the two-dimensional Gaussian distribution and the area outside the edge with a confidence of 0.1. The calculation is the same as the size of the scene, and
each pixel has a confidence distribution map of its corresponding feature points; that is, a confidence attribute prob$_{i,j}$ is assigned to
each point prob$_{i,j}$

$$\text{prob}_{i,j} = \frac{1}{2\pi \sigma_x \sigma_y} \exp\left[-\frac{1}{2} \left( \frac{X - u}{\sigma_x} \right)^2 \right], \text{inboundingbox}$$
$$0.1, \text{outboundingbox}$$

Among them, $\sigma$ is the covariance matrix describing the correlation between variables, $d$ is the variable dimension, $u$ is the center
coordinate vector of the semantic segmentation box, and $X = (i, j)$
is the coordinate vector of the point $\text{prob}_{i,j}$. $i$ and $j$ are the $x$ and $y$-axis coordinates of the pixel’s position in the screen, respectively, so prob$_{i,j}$ is calculated according to the pixel’s position on the image. It is possible that suppose the same pixel appears in multiple
of the bounding box, according to our comparative experiments. In
that case, we finally select the maximum value of the probability values calculated in each bounding box as the probability value of
the point.

2.3 BA problem algorithm based on confidence
distribution graph
In the SLAM process, Xiang Gao et al. [21] proposed that, it is
usually necessary to infer the pose and positioning according to
the noise due to various interference and misrecognition noise.
According to the confidence distribution map, we assign, Attribute
prob$_{i,j}$, confidence to each pixel of each frame of the picture. When
the feature point extraction is completed, and we can optimize the
back end according to the confidence attribute of the feature point
in Figure 2. Since SLAM uses G2O optimization to solve the Bundle
Adjustment problem, we multiply the probability attribute into the
covariance matrix corresponding to each error term of the least
squares problem constructed by the Bundle Adjustment problem
to introduce the probability attribute to each error term. Weights
we control the weight of the edge to achieve the same purpose of achieving the straight line. This experiment is based on the ROS system for data transmission.

3.2 Probability graph thread

Bounding box's center position and side length information make target detection a crucial auxiliary tool for SLAM positioning. To alleviate the impact of the target detection network on the timeliness of trying SLAM as much as possible, we choose YOLOv5 for target detection. We determine object detection optimizes the weight of the approximate position of the object.

\[
T^* = \arg\min \frac{1}{2} \sum_{k=1}^{n} \text{prob}_k \times ||U_k - \frac{1}{s_k} KTP_k||^2_2
\]

\(T\) is the Lie group of the camera pose \(R, U_k = \{u_k, v_k\}^T\) is the projection coordinate of a particular spatial point \(P_k = \{X_k, Y_k, Z_k\}\), \(K\) is the camera internal parameter matrix, in the g2o optimizer, we control the weight of the edge to achieve the same purpose of controlling the weight of the error term, to realize the prob-based BA problem-solving method.

3 SPECIFIC IMPLEMENTATION

3.1 Target detection optimizes

This experiment is based on the ROS system for data transmission. In our scheme, on the one hand, a lightweight network, yolov5, is used for semantic segmentation. On the other hand, the positioning function of the target detection network is used to obtain information related to the target recognition object. For example, the bounding box’s center coordinate information and the bounding box’s side length information make target detection a crucial auxiliary tool for SLAM positioning. To alleviate the impact of the target detection network on the timeliness of trying SLAM as much as possible, we choose YOLOv5 for target detection. We determine that the set of dynamic and static objects that may appear in the scene according to the environment and prerequisites. After obtaining the recognition result, based on this target recognition, we perform semantic segmentation on the original image and finally obtain the bounding box’s center position and side length information.

3.2 Probability graph thread

Object detection techniques have great advantages in real-time performance compared with purely semantic segmentation methods but cannot provide accurate object masks. In the indoor dynamic SLAM scene, by setting up a probability distribution map module after object detection, we reduce the influence of these feature points in the subsequent process by assigning low confidence to each feature point distributed in the non-target recognition object. At the same time, according to the two-dimensional Gaussian distribution formula, the confidence of the feature points in each bounding box is defined for the bounding box information obtained in the semantic segmentation, and the attention to the particular static objects existing in the environment is strengthened. The distribution is distributed on these particular static objects. The feature points are more involved in optimization and pose estimation, which enhances positioning accuracy. At the same time, the two-dimensional Gaussian distribution is used because it can reduce the negative impact of inaccurate semantic segmentation as much as possible, thereby improving the effect of camera pose estimation, improving the accuracy of camera pose estimation, and reducing the amount of computation. Finally, we construct a confidence matrix and transmit it as a new topic according to the confidence prob attribute obtained from the two-dimensional Gaussian distribution.

3.3 Optimization in G2O

After ORB-SLAM2 normally processes the lens image information from the monocular camera, we combine the probability distribution map with the original static SLAM and multiply the probability attributes of each positioning point into the covariance matrix corresponding to each error term constructed by the Bundle Adjustment problem.

4 EXPERIMENTAL EVALUATION

In order to compare the effects of PROB-SLAM more effectively, this paper compares ORB-SLAM3 and Dyna-SLAM, two representative and widely used SLAM systems today, in the same scenario, and measures the effects of PROB-SLAM from multiple evaluation indicators, we use TUM’S RGB-D dataset [22], which has more dynamic objects, to test the performance optimization of our proposed algorithm for SLAM that introduces semantic segmentation methods. The evaluation indicators used to evaluate the accuracy are absolute trajectory error (ATE) and relative pose error (RPE). ATE stands for global consistency of trajectories. RPE includes translation drift and rotation drift. The root means square error (RMSE) and standard deviation (S.D.) of both are used to represent the robustness and stability of the system [23].

We first show the impact of the PROB algorithm on the positioning points of the original SLAM to detect whether the PROB algorithm will cause too much performance impact on the original ORB-SLAM2[5] and then compare our method with the original ORB-SLAM2 running map trajectory in Figure 3, to determine whether the error has been reduced, to achieve the optimal effect, and then design a series of ablation experiments to test the influence of the PROB module, as well as the natural effect of the probability map concept created by running it alone, and finally conduct real-time analysis and calculation. All experiments were performed on a computer with an Intel i7 CPU, Nvidia 1070 GPU, and 16GB of memory.
Figure 3: The level of feature points obtained in PROB-SLAM is consistent with that of the original SLAM, and there is no reduction in the number of feature points. Thus, the addition of target detection and the corresponding probability map construction thread has little impact on SLAM performance.

Figure 4: The result is processed by the PROB algorithm. We selected several locations where traditional SLAM is prone to significant errors for effect display. Due to the presence of dynamic objects and some unclear fields of view, the original SLAM has significant errors. The experimental results show that the PROB algorithm can effectively alleviate the error problem in these scenarios, and the specific data will be shown later.

4.1 Anchor point performance comparison experiment

In the ORB-SLAM operation scene, the motion of the object, the unclear object to be detected in the image presented in the camera’s field of view, the blurred image caused by the rotation of the camera, and the singular angle of view all bring severe challenges to SLAM positioning, and it is easy to locate errors. If it increases, it will even lead to the loss of positioning. The following Figure 4 is a typical situation.

It is worth noting that there is no performance loss in the scene where many objects are recognized for target detection, and the performance is lower than the original version—many burdens and impacts. However, static SLAM cannot discriminate whether the character is moving, resulting in a misjudgment of critical points’ dynamic and static attributes. We use the PROB algorithm to optimize critical points based on the target detection probability map, which can well avoid this problem in Figure 5. Through the processing of the PROB algorithm, experiments have proved that the error, in this case, is significantly reduced according to the data response of multiple evaluation indicators. The experimental results fully demonstrate the effectiveness and robustness of the PROB algorithm.

4.2 Running error comparison

Our experimental results with ORB-SLAM2 show that, unlike the PROB-SLAM algorithm, which has advantages over ORB-SLAM2 in both dynamic and static scenes, the algorithm can achieve a specific improvement in both dynamic and static scenes. The APE (Figure 6) and RPE (Figure 7) graphs of our algorithm are as follows:

The Figure 7 shows the comparison between the original ORB-SLAM2 and the modified version, where POSE is the version with the probability map added, and POSE-DYNAMIC uses YOLO to remove dynamic objects for predefined knowledge, which shows that in the range of error, it will be improved compared to traditional visual SLAM.

4.3 Separate module experiments

To prove the function of our PROB algorithm module, we designed a slice experiment, among them, PROB-SLAM: the algorithm of this paper, Only-PROB: direct target detection and convert the target detection image information into a probability map form.

The results show that the PROB algorithm can directly improve the confidence of the position of crucial points for static objects and can successfully adjust the influence of dynamic objects. It can play a perfect optimization effect in the process of object movement. If Only-PROB adjusts the contrast adjustment index of the probability map too large, it will also cause the loss of positioning.
Table 1: Processing time of each module.

| Methods      | YOLO | PROB | Tracking |
|--------------|------|------|----------|
| ORB-SLAM2    | /    |      | 32.5ms   |
| PROB-SLAM    | 12.28ms | 0.12ms | 36.5ms   |
| Dyna-SLAM    | /    |      | 39.6ms   |
| ORB-SLAM3    | /    |      | 27.3ms   |

Figure 6: In the line chart, the four lines represent four SLAM systems respectively. When ORB-SLAM2 is at the highest error point, PROB-SLAM is relatively less than about 15% of its error, while compared with Dyna-SLAM, the maximum error is about 8% more accurate. In the second half of the dataset, there are more dynamic objects in the scene, and PROB-SLAM gradually has an advantage over ORB-SLAM3 in terms of error of about 5%.

Figure 7: In terms of relative error, the relative error distribution position of PROB-SLAM is also relatively lower than ORB-SLAM2, which is in the middle of the four systems.

4.4 Processing time comparison

Real-time performance is a significant indicator of the SLAM system, and a more considerable delay will cause a more significant error in SLAM positioning. We tested the average running time of each module, as shown in the Table 1. It includes the time required for ORB-SLAM2 original processing, the time required for PROB-SLAM processing, and the time required for image processing by a separate PROB algorithm. YOLOv5-based semantic threading runs in parallel with ORB feature extraction. The results show that the average processing time per frame of the PROB-SLAM main thread is:

5 CONCLUSION

This paper proposes an independent thread for constructing a probability distribution map based on semantic information, which can reduce the impact of semantic information's uncertainty. Because it is an independent thread, it can be easily integrated into various existing SLAMs. In the system, it will not affect the performance of the original system, so it has a wide range of application scenarios. For example, in some industrial scenarios with fixed particular objects, we can introduce this thread into the existing SLAM system. Accuracy is improved by better utilization of semantic detection information.

REFERENCES

[1] N. Axel, J. Christian, E. Rayrarmoglu, and O. Rav, "Visual Navigation for Mobile Robots," InTech, 2010. doi: 10.5772/9292.
[2] S. Yang, S. A. Scherer, X. Yi, and A. Zell, "Multi-camera visual SLAM for autonomous navigation of micro aerial vehicles," Robotics and Autonomous Systems, vol. 93, pp. 116–134, 2017. doi: 10.1016/j.robot.2017.03.018.
[3] Z. Zhang, M. Shu, Z. Wang, H. Wang, and X. Wang, "A Registration Method for Augmented Reality System Based on Visual SLAM," 2019. doi: 10.1109/icimag.2019.00094.
[4] R. Mur-Artal, J. M. Montiel, and J. D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," IEEE Trans. Robot., vol. 31, no. 5, pp. 1147–1163, 2015, doi: 10.1109/tro.2015.2463671.
[5] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," IEEE Trans. Robot., vol. 33, no. 5, pp. 1255–1262, 2017, doi: 10.1109/tro.2017.2705103.
[6] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," Cham: Springer International Publishing, 2014, pp. 834–849. doi: 10.1007/978-3-319-10605-2_54.
[7] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems," IEEE Trans. Robot., vol. 33, no. 2, pp. 249–265, 2017. doi: 10.1109/tro.2016.2623335.
[8] C. Cadena et al., "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," IEEE Trans. Robot., vol. 32, no. 6, pp. 1309–1332, 2016. doi: 10.1109/tro.2016.2624754.
[9] W. Dai, Y. Zhang, P. Li, Z. Fang, and S. Scherer, "RGB-D SLAM in Dynamic Environments Using Point Correlations," IEEE Trans. Pattern Anal. Mach. Intell., pp. 1–1, 2020, doi: 10.1109/tpami.2020.3010942.
[10] D.-H. Kim and J.-H. Kim, "Effective Background Model-Based RGB-D Dense Visual Odometry in a Dynamic Environment," IEEE Trans. Robot., vol. 32, no. 6, pp. 1565–1573, 2016. doi: 10.1109/tro.2016.2609395.
[11] S. Li and D. Lee, "RGB-D SLAM in Dynamic Environments Using Static Point Weighting," IEEE Robot. Autom. Lett., vol. 2, no. 4, pp. 2263–2270, 2017, doi: 10.1109/lra.2017.2724759.
[12] T. Zhang, H. Zhang, Y. Li, Y. Nakamura, and L. Zhang, "FlowFusion: Dynamic Dense RGB-D SLAM Based on Optical Flow," 2020. doi: 10.1109/icra40945.2020.9197349.
[13] K.-N. Lianos, J. L Schönberger, M. Pollefeys, and T. Sattler, "VSO: Visual Semantic Odometry," Cham: Springer International Publishing, 2018, pp. 246–263. doi: 10.1007/978-3-030-01225-0_15.
[14] Y. Liu, H. Gao, L. Guo, A. Qin, C. Cai, and Z. You, "A Data-Flow Oriented Deep Ensemble Learning Method for Real-Time Surface Defect Inspection," IEEE Trans. Instrum. Meas., vol. 69, no. 7, pp. 4681–4691, 2020. doi: 10.1109/tim.2019.2907849.
[15] F. Li et al., "A Mobile Robot Visual SLAM System With Enhanced Semantics Segmentation," IEEE Access, vol. 8, pp. 25442–25458, 2020, doi: 10.1109/access.2020.2970238.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016. doi: 10.1109/cvpr.2016.91.

[17] Y. Liu and J. Miura, "RDS-SLAM: Real-Time Dynamic SLAM Using Semantic Segmentation Methods," IEEE Access, vol. 9, pp. 23772–23785, 2021, doi: 10.1109/access.2021.3050617.

[18] L. Guo, Y. Lei, N. Li, T. Yan, and N. Li, "Machinery health indicator construction based on convolutional neural networks considering trend burr," Neurocomputing, vol. 292, pp. 142–150, 2018, doi: 10.1016/j.neucom.2018.02.083.

[19] L. Zhao, Z. Liu, J. Chen, W. Cai, W. Wang, and L. Zeng, "A Compatible Framework for RGB-D SLAM in Dynamic Scenes," IEEE Access, vol. 7, pp. 75604–75614, 2019, doi: 10.1109/access.2019.2922733.

[20] D. Yang, S. Bi, W. Wang, C. Yuan, X. Qi, and Y. Cai, "DRE-SLAM: Dynamic RGB-D Encoder SLAM for a Differential-Drive Robot," Remote Sensing, vol. 11, no. 4, p. 380, 2019, doi: 10.3390/rs11040380.

[21] X. Gao, T. Zhang, Y. Liu, Q. Yan. 2017. 14 Lectures on Visual SLAM: From Theory to Practice. Publishing House of Electronics Industry.

[22] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," 2012. doi: 10.1109/iros.2012.6385773.

[23] Y. Fan, Q. Zhang, Y. Tang, S. Liu, and H. Han, "Blitz-SLAM: A semantic SLAM in dynamic environments," Pattern Recognition, vol. 121, p. 108225, 2022, doi: 10.1016/j.patcog.2021.108225.