IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 8 November 2022, accepted 5 December 2022, date of publication 9 December 2022,
date of current version 15 December 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3228040*

RESEARCH ARTICLE

# STDC-SLAM: A Real-Time Semantic SLAM Detect Object by Short-Term Dense Concatenate Network

**ZHANGFANG HU, JIAN CHEN, YUAN LUO, AND YI ZHANG**
Key Laboratory of Optoelectronic Information Sensing and Technology, School of Advanced Manufacturing Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Corresponding author: Jian Chen (782090654@qq.com)

**ABSTRACT** Visual Simultaneous Localization and Mapping (SLAM) plays an important role in computer vision and robotic field. With the development of Convolutional Neural Network (CNN), most scholars currently fuse CNN with visual SLAM to reduce the impact of dynamic objects on visual SLAM. To address the impact of semantic segmentation networks with lower precision and quad-tree algorithm with over-uniform distribution of feature points on the location accuracy of SLAM, we proposed an STDC-SLAM: Short-Term Dense Concatenate Network SLAM, which was based on ORB-SLAM3. In the proposed system, a real-time STDC network was used for semantic thread to segment dynamic objects. On the one hand, we designed a segmentation refinement module to optimize the semantic segmentation maps using images depth information. On the other hand, we improved the Qtree-ORB algorithm by reducing the iterations of low-quality feature points in the rejection thread. We have evaluated our SLAM in public data sheets and compared it with ORB-SLAM3, DynaSLAM, PSPNet-SLAM. Experiments showed that our SLAM improved in localization accuracy compared to DynaSLAM and in processing speed compared to DynaSLAM and PSPNet-SLAM.

**INDEX TERMS** STDC-SLAM, dynamic, semantic segmentation, Qtree-ORB algorithm.

## I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) is an algorithm that enables a mobile robot to simultaneously locate its own position and construct a map of its surroundings without a priori information [1], [2]. SLAM is mainly used in mobile robots, driverless cars, Augmented Reality (AR). Traditional SLAM has extremely high static requirements for the environment. And excellent performance can only be achieved in ideal static environments, such as ORB-SLAM2 [3], ORB-SLAM3 [4], MonoSLAM [5], LSD-SLAM [6]. In particular, ORB-SLAM3 uses the Oriented Fast and Rotated Brief (ORB) [7] algorithm, which has superior performance. Therefore ORB-SLAM3 is used as the base framework of our SLAM.

Although visual SLAM has made great progress, there are still some problems that need to be solved. For example, the robustness of SLAM can be seriously affected by dynamic objects when mobile robots work in dynamic environments. Also, many existing visual SLAM algorithms assume the environment to be static, ignoring the influence of dynamic objects [8], [9]. The application scope of traditional SLAM is limited by the dynamic objects in the real environment. To apply SLAM in dynamic environments, many scholars have combined Convolutional Neural Network (CNN) with traditional SLAM, such as PSPNet-SLAM [10], Blitz-SLAM [11], DS-SLAM [12]. SLAM fused with CNN can achieve semantic understanding of the environment. Meanwhile, the influence of dynamic targets on SLAM can be removed using CNN.

The accuracy and robustness of SLAM in dynamic environments are improved with the incorporation of CNN. However, the time-consuming CNN limits the application of SLAM. With the development of CNN, more and more excellent networks are proposed, such as SegNet [13], PSPNet [14], BiSeNet [15], STDC [16]. Compared with other semantic segmentation networks, Short-Term Dense Concatenate (STDC) network uses detail guidance module to improve

The associate editor coordinating the review of this manuscript and approving it for publication was Sudipta Roy.

segmentation accuracy. And the detail guidance module is removed in the inference phase to speed up the processing speed of the network.

In this paper, we propose a real-time semantic SLAM system which selects the STDC network as the semantic segmentation network in semantic thread. The system based on the ORB-SLAM3 algorithm framework. In the semantic thread, we use the STDC network to obtain the segmentation map. Meanwhile, we design a segmentation refinement module to optimize the segmentation map. This module combines the segmentation map with depth information of the same frame. In the rejection thread, we improve the Qtree-ORB algorithm in ORB-SLAM3 to further improve the location accuracy of the system.

In summary, we highlight our main contribution below:

- We proposed the algorithm framework of STDC-SLAM, and introduced the STDC network as a semantic thread on the basis of ORB-SLAM3. The network can efficiently segment dynamic objects by detail guidance. The detail guidance is used only in the inference phase, so that the network can segment dynamic objects in continuous frames more quickly and reliably.
- A segmentation refinement module is proposed in the semantic thread. This module combines the segmentation map with the depth information of the same frame to optimize this segmentation map, and improve the segmentation accuracy of the system for dynamic objects.
- In the rejection thread, we improve the rejection module of ORB-SLAM3. On the one hand, this module can reject dynamic feature points using segmentation maps. On the other hand, the Qtree-ORB algorithm is improved to reject redundant feature points and retain more high-quality feature points.

In the rest of this paper, we firstly discuss the related work. Next, our proposed system is described in detail. Then, the effectiveness of the system is proved with experimental results. Finally, we conclude and discuss the paper.

## II. RELATED WORK

SLAM is divided into laser SLAM and visual SLAM according to the class of sensors. Visual SLAM is widely applied compared to laser sensors because it can obtain more environmental information by using cameras. With the development of visual SLAM, it is further divided into traditional visual SLAM and dynamic visual SLAM. Traditional visual SLAM strongly assumes the environment as static for the convenience of the algorithm, such as MonoSLAM [5], LSD-SLAM [6], ORB-SLAM [17]. MonoSLAM creates a sparse map online using a probabilistic statistical framework and uses the Extended Kalman Filter (EKF) algorithm to optimize the camera position. The algorithm has good performance by randomly selecting search frames for feature points extraction and using image blocks for feature matching. However, the extracted sparse features are unstable and easily lead to tracking failure. Klein and Murray [18] propose a PTAM algorithm that firstly realized the parallel processing of tracking and map building. This algorithm is a keyframe-based monocular SLAM, which only needs to optimize the key images when performing camera pose optimization. In addition, PTAM separates the front and back ends and proposes a nonlinear optimization method. It is important for the development of subsequent visual SLAM. In 2015, Mur-Artal et al. propose ORB-SLAM based on PTAM. ORB-SLAM uses feature points extracted by ORB algorithm for feature matching. The system shows higher localization accuracy in tests compared to PTAM and can work in real time on the CPU. ORB-SLAM has better performance compared to other traditional visual SLAM. Subsequently, the authors have successively proposed ORB-SLAM2 [3] and ORB-SLAM3 [4]. Traditional visual SLAM can achieve real-time and stability in static environments, but its strong assumptions limit its application scope.

Since the real environment is often accompanied by a large number of dynamic objects, visual SLAM needs to identify and reject dynamic objects. So that, the visual SLAM only tracks the static environment and constructs the static environment map. To identify dynamic targets in the environment, Kundu et al. [19] use the fundamental matrix to estimate the distance between the matching feature and the epipolar line in two adjacent frames. When the distance reached a predetermined threshold, the object was considered as a dynamic one. This method uses a geometric approach to identify dynamic targets in dynamic environments, but its accuracy is low. Wang et al. [20] identify independent moving objects in the scene by the polar geometric constraints of matched points in two adjacent frames and the clustering information of the depth map provided by the RGB-D camera. However, this method depends on the positional transformation matrix between two adjacent frames, and its error is large in highly dynamic scenes. Y. Fan et al. [21] propose two mathematical geometric constraints to localize the dynamic regions in the scene. Firstly, the information of the dynamic regions is set as blank. Then one of the image sequences is used as the projection plane. Finally, the fused image is obtained by projecting other image sequences to this plane. However, the robustness of the SLAM in dynamic scenes is not well solved by geometric methods alone. Fang and Dai [22] detects and removes dynamic objects using the optical flow method, which improves the accuracy of the positional estimation. However, this method has limited improvement on the overall performance of the SLAM system. Bakkay et al. [23] uses a scene flow method based on optical flow method to detect dynamic objects and uses a region growing algorithm to separate dynamic and static objects. This method improves the accuracy of feature points matching. However, the real-time performance is poor.

In recent years, CNN have been widely used in computer vision. Its parameters are trained by a large amount of data to make pixel-level prediction of images. For example, SegNet [13], PSPNet [14], BiSeNet [15], STDC [16]. They are trained with datasets to perform semantic recognition

of the environment, such as identifying people, cars, animals, sidewalks, etc. Combining CNN with SLAM has been a popular research direction to solve the problems of SLAM in dynamic environments [24], [25] by using CNN to understand the environments semantically, such as PSPNet-SLAM [10], Blitz-SLAM [11], DS-SLAM [12], DynaSLAM [26], DDL-SLAM [27] and so on. PSPNet network is used in PSPNet-SLAM to perform semantic segmentation of dynamic objects in the scene. In addition, an optimal error-compensated single-response matrix is designed in the geometric thread to improve the accuracy of dynamic points detection. DS-SLAM adds an independent thread to perform SegNet semantic segmentation and obtain semantic information. And the tracking thread performs feature extraction and moving consistency detection. Then the tracking thread uses the semantic information from the semantic thread to perform dynamic feature points rejection. In DynaSLAM, the combination of multi-view geometry and Mask RCNN is used to detect and filter dynamic objects, which reduces the localization accuracy of SLAM and improves the robustness of the SLAM system. Ai and Rui et al. propose a DDL-SLAM using DUNet [28] network for semantic segmentation and combining multi-view geometry to detect dynamic objects, which greatly improves the localization accuracy of SLAM. These SLAM use various CNN to recognize the environment and reject dynamic objects in the environment. These methods improve the robustness, accuracy and stability of SLAM systems. However, the processing speed of segmentation is required to fulfill the real-time performance to enable SLAM to be applied in real-world applications, such as driverless cars and mobile robots.

Feature-based visual SLAM optimizes the position and the 3D coordinates of feature points by extracting feature points of each frame for feature points matching. The main feature points extraction algorithms are Scale Invariant Feature Transform (SIFT) [29], Speed-UP Robust Feature (SUFER) [30], ORB, etc. The SIFT algorithm has good scale invariance and rotation invariance, but its running time is long. SUFER is developed based on SIFT, which greatly improves the processing speed compared to SIFT, but its real-time performance is still not excellent because it is time consuming to compute its feature descriptors. The ORB algorithm is proposed based on Features from Accelerated Segment Test (FAST) [31] and Binary Robust Independent Elementary Features (BRIEF) [32], and solves the problem that the FAST algorithm does not have rotational invariance by adding an orientation module to the FAST feature points. To address the problem of non-uniform distribution of feature points in the standard ORB algorithm, Raúl et al. propose an ORB-SLAM algorithm that uses a quadtree to homogenize the distribution of feature points and also adopts an adaptive threshold method to improve the ORB algorithm. Although this method improves the uniformity of feature points distribution, it still has the problem of too many iterations and over-uniformity. Moreover, this method extracts some low-quality feature points, which reduces the localization accuracy of SLAM. Sun et al. [33] propose a multi-probe based A-ORB algorithm. Although this algorithm has high robustness, the number of extracted feature points is small and the uniformity of feature points distribution is poor. Li et al. [34] use a multi-task feature extraction network to extract feature points in a scene by combining SLAM with a deep learning approach. Although this algorithm has significant advantages over traditional feature points extraction methods in complex scenes, it sacrifices real-time performance and relies on GPU acceleration.

In this paper, we propose an STDC-SLAM that adds a parallel semantic thread to the ORB-SLAM3 algorithm. This thread uses a real-time semantic segmentation network for semantic recognition of the environment and removes the influence of dynamic objects on the SLAM algorithm. In the semantic thread, we use the STDC network as the semantic segmentation network. In order to improve the segmentation accuracy of the system for dynamic objects, we propose a segmentation refinement module that combines the original segmentation map with the depth information of the corresponding image to obtain a new segmentation map with higher segmentation accuracy. In the rejection thread, we have improved the rejection module. We use the optimized segmentation map to eliminate the feature points on the dynamic objects. Then, the redundant feature points are removed from the remaining static feature points by the improved Qtree-ORB algorithm. This improved algorithm improves the overall quality of feature points by reducing the iterations of low-quality feature points. On the one hand, a sufficient number of high-quality feature points are retained after the dynamic feature points are removed by the corresponding segmentation map. On the other hand, the remaining feature points satisfy the uniformity requirement. In total, our system combines STDC network, segmentation refinement module and improved Qtree-ORB algorithm into ORB-SLAM3, showing its excellent robustness and real-time performance.

## III. SYSTEM DESCRIPTION

Figure 1 gives an overview of our system. Our system added a parallel semantic thread based on ORB-SLAM3. Firstly, the RGB information and depth information of images are imported from the RGB-D camera. Secondly, the extraction thread extracts the feature points of the input image. Meanwhile, the semantic thread segments the input image by the pre-trained STDC network. Then, the segmentation refinement module combines the segmentation map with the depth information of the corresponding image. Finally, the rejection thread uses the feature points map and the optimized segmentation map to reject dynamic feature points.

### A. SEMANTIC SEGMENTATION STDC NETWORK

STDC networks are used as a semantic segmentation network for semantic understanding and segmentation of dynamic objects in realistic environments. The STDC network framework is given in Figure 2. The network is composed of

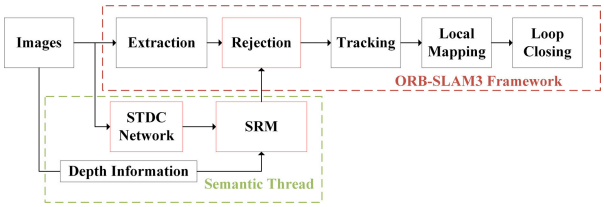


**FIGURE 1. Overview of the STDC-SLAM system. The system is built on the ORB-SLAM3 framework, and we propose a semantic thread. Images contain RGB information and depth information, and SRM denotes Segmentation Refinement Module.**

three main parts: Network Architecture, Training Loss, and Detail Ground-truth Generation. The Network Architecture is shown in Figure 2(a). It consists of five stages and one pooling layer. Each stage is composed of several STDC modules [10], shown in Figure 3(a). Figures 3(b) and 3(c) show the structure of the STDC module, which fuses feature maps from four different connectivity layers. The structure preserves scalable respective fields and multi-scale information. The STDC network uses Attention Refine Module (ARM) to obtain context information. Meanwhile, the Feature Fusion Module (FFM) is used to combine the context information and spatial information to obtain the segmentation prediction.

To improve the segmentation accuracy, STDC network propose a Detail Guidance module to guide the low-level layers to learn the spatial information as shown in Figure 2(b). First, the detail map ground-truth is generated from the segmentation ground-truth by Laplacian operator as shown in Figure 2(c). As illustrated in Figure 2(a), the Detail Head is inserted in Stage 3 to generate the detail feature map. Then, the detail ground-truth is used as the guidance of detail feature map to guide the low-level layers to learn the feature of spatial details. Finally, the learned detail features are fused with the context features from the deep block of the decoder for segmentation prediction. Note that the Training Loss and the Detail Ground-truth Generation are discarded in the inference phase. Therefore, STDC network has a higher segmentation accuracy and achieves real-time performance for segmentation task.

### B. SEGMENTATION REFINEMENT MODULE

We use the STDC network as a semantic segmentation network to segment out dynamic objects in the environment. However, the network still has the problem of incomplete segmentation of dynamic objects. This problem affects the accuracy and stability of SLAM. To solve this problem, we propose a Segmentation Refinement Module (SRM)that combines the segmentation map with the depth information of the same frame to improve the segmentation accuracy.

We divide the classes segmented by STDC network into two main classes, static class and dynamic class. To reduce the influence of dynamic feature points on the stability of the SLAM system, as well as to consider the real-time performance of the SLAM system, we select the direct calibration method. This method artificially sets the highly dynamic objects in the environment as dynamic objects. For example,

**Algorithm 1** Segmentation Refinement Algorithm

**Input:** Depth image $I_D$,Original dynamic set $D^o$;
**Output:** New dynamic set $D^n$;

```
for (u_i,v_i) in D^o do
    d_i = I_D(u_i,v_i);
    Mask = {(u_i-8,v_i-8),...,(u_i,v_i),...,(u_i+8,v_i+8)};
    for (x_j,y_j) in Mask do
        d_j = I_D(x_j,y_j);
        if |d_i − d_j| ≤ τ then
            Insert((x_j,y_j),D^n);
        end if
    end for
end for
```

people, animals, cars, etc. Other objects are set as static objects. The dynamic class set $D^o$ includes all dynamic pixel point coordinates $\{(u_1,v_1),\ldots,(u_k,v_k)\}$.

Based on the continuity of depth of the objects, we reclassify the static pixel points around each dynamic pixel point according to the depth difference. Firstly, the dynamic pixel point $(u_i,v_i)(1\leq i\leq k)$ is taken as the center point and the depth value $d_i$ of this pixel point is recorded. Then, the coordinates of 389 pixel points $\{(x_1,y_1),\ldots,(x_{389},y_{389})\}$ are determined using a mask of size $17\times17$. Finally, if the difference between the depth value $d_j(1\leq j\leq389)$ of the static pixel point $(x_j,y_j)$ and the depth value $d_i$ of the dynamic pixel point $(u_i,v_i)$ is less than or equal to the threshold $\tau$, the pixel point is added to the new dynamic class set $D^n$, as follows:

$$\begin{cases} |d_i - d_j| \leq \tau & (x_j, y_j) \in D^n \\ else & (x_j, y_j) \notin D^n \end{cases} \tag{1}$$

where $\tau$ is a preset threshold value, and we set $\tau = 500$ (0.1m), according to Blitz-SLAM[6]. The segmentation refinement algorithm is shown in Algorithm 1. The algorithm takes the depth image $I_D$ and the original dynamic set $D^o$ as input. Firstly, the depth value $d_i$ of the coordinate point $(u_i,v_i)$ in $D^o$ is calculated by $I_D(u_i, v_i)$, and a *Mask* of size $17 \times 17$ containing 389 coordinate points centered on the point $(u_i,v_i)$ is obtained, such as $\{(u_i-8,v_i-8),\ldots,(u_i,v_i),\ldots,(u_i+8,v_i+8)\}$. Then the depth value $d_j$ of coordinate points $(x_j,y_j)$ in *Mask* is calculated by $I_D(x_j, y_j)$. If the difference between $d_j$ and $d_i$ is less than or equal to $\tau$, $(x_j,y_j)$ is inserted into the new dynamic set $D^n$.

### C. IMPROVED QTREE-ORB ALGORITHM

In the ORB-SLAM3 system, the ORB algorithm is used in the extraction thread to extract feature points from the input frames, and the quad-tree algorithm is used in the rejection thread to reject redundant feature points in the feature maps. These two algorithms are collectively called the Qtree-ORB algorithm. By this algorithm, the remaining feature points can be uniformly distributed in the feature maps. However, the Qtree-ORB algorithm has suffered from over-uniform distribution of feature points and retaining a large number of low-quality feature points. These problems reduce the accuracy of pose estimation in SLAM system.

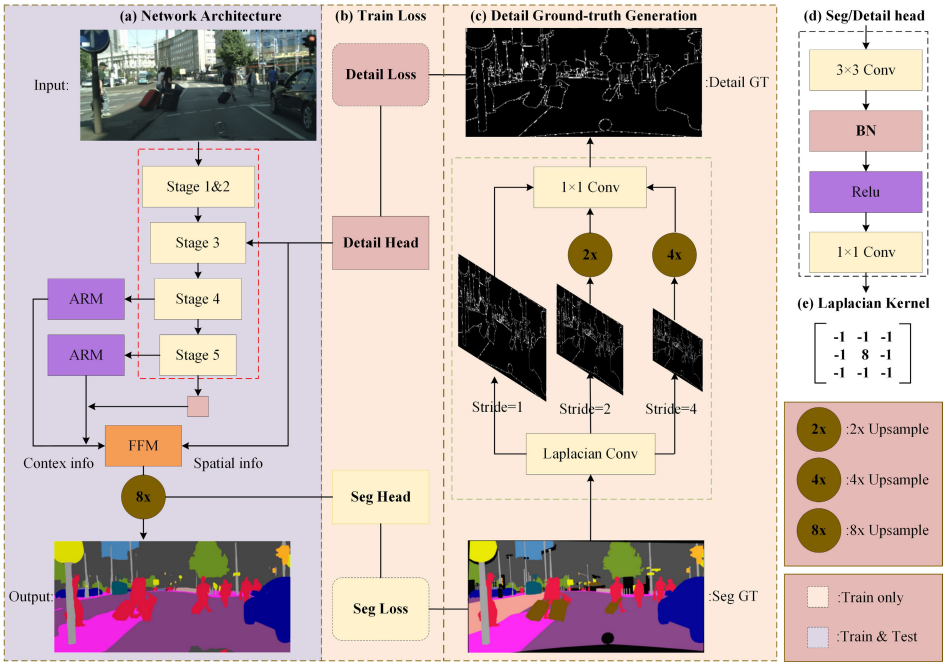


**FIGURE 2.** Overview of the STDC Segmentation network. ARM denotes Attention Refine module, and FFM denotes Feature Fusion Module in [10].

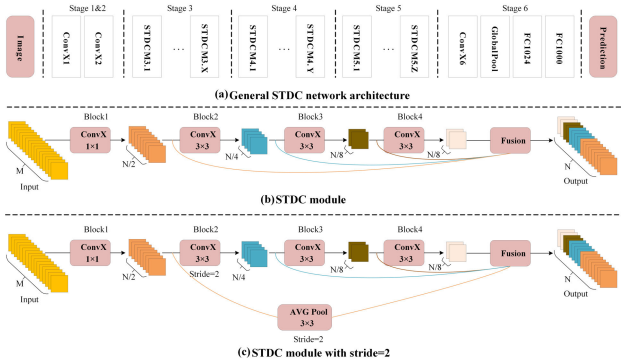


**FIGURE 3.** (a) General STDC network architecture. ConvX operation refers to the Conv-BN-ReLU. (b) Short-Term Dense Concatenate module (STDC module). M denotes the dimension of input channels, N denotes the dimension of output channels. Each block is a ConvX operation with different kernel size. (c) STDC module with stride=2.

In this paper, we propose an improved Qtree-ORB algorithm to improve the overall quality of feature points by reducing the iterations of low-quality feature points.

Before extracting the feature points, we construct the image pyramid by dividing the input image into $M$ layers considering the scale transformation of the feature points, as follows:

$$M = \text{Round}[\text{Max}(W, H)/100] \tag{2}$$

where $W$, $H$ denote the width and height of the input image respectively, and Round denotes the rounding function, Max denotes the maximum value function. The total area $S$ of the image pyramid is calculated by Equation 3, as follows:

$$\begin{aligned} S &= HW(s^2)^0 + HW(s^2)^1 + \ldots + HW(s^2)^{(M-1)} \\ &= HW\frac{1-(s^2)^M}{1-s^2} = C\frac{1-(s^2)^M}{1-s^2} \end{aligned} \tag{3}$$

where $s$ denotes the preset scaling factor, and $C$ denotes the original image area of the first layer.

To distribute the feature points uniformly on the feature map, the number of feature points per unit area is set to $X$ obtained by Equation 4:

$$X = \frac{N}{S} = \frac{N}{C\frac{1-(s^2)^M}{1-s^2}} = \frac{N(1-s^2)}{C(1-(s^2)^M)} \tag{4}$$

where $N$ denotes the total number of feature points. The number of feature points $N_i$ in the $i$-th $(0 \le i \le (M-1))$ layer is calculated by Equation 5, as follows:

$$N_i = \frac{N(1-s^2)}{C(1-(s^2)^M)}C(s^2)^i = \frac{N(1-s^2)}{1-(s^2)^M}(s^2)^i \tag{5}$$

The feature points extraction algorithm is shown in Algorithm 2. Feature points in each layer of the image pyramid are extracted by Algorithm 2. Firstly, the $i$-th layer of the pyramid images $L_i$ is divided into multiple regions of equal size, such as $\{A_1, \ldots, A_L\}$. Then, the Features From Accelerated Segment Test (FAST) algorithm with different thresholds is used to extract the feature points. The set of high-quality feature points $P^H$ is extracted separately for each region by the FAST algorithm with high threshold $T_H$, such as FAST($A_j$,$T_H$). If the total number of extracted high-quality feature points $N^H$ is greater than or equal to the number of feature points in $i$-th layer $N_i$, the extraction of feature points in this layer is ended. Conversely, the set of low-quality feature points $P^L$ is extracted for each region separately using the FAST algorithm with low threshold $T_L$. When the number of all feature points $N^S$ is greater than or equal to $N_i$, the extraction of feature points in this layer is ended. According to ORB-SLAM3, we set $T_H$=20 and $T_L$=7.

**Algorithm 2** Feature Points Extraction Algorithm

**Input:** $I$-th layer of the image pyramid $L_i$,
Number of feature points in $i$-th layer $N_i$;
**Output:** High-quality points set $P^H$,
Low-quality points set $P^L$;

$\{A_1, \ldots, A_L\}$ in $L_i$;
**for** $A_j$ in $L_i$ **do**
  $P_j^H$ = FAST($A_j$,$T_H$);
  Insert($P_j^H$,$P^H$);
**end for**
$N^H$ = CalculatePointsNumber($P^H$);
**if** $N^H \leq N_i$ **then**
  **for** $A_j$ in $L_i$ **do**
    $P_j^L$ = FAST($A_j$, $T_L$);
    Insert($P_j^L$,$P^L$);
    $N^L$ = CalculatePointsNumber($P^L$);
    $N^S = N^H + N^L$;
    **if** $N^S \geq N_i$ **then**
      break;
    **end if**
  **end for**
**end if**

**Algorithm 3** Improved Quad-Tree Algorithm

**Input:** $I$-th layer $L_i$, High-quality points set $P^H$;
**Output:** Retained feature points set $P^R$;

*Node* = InitializeNodesSet($L_i$);
$N^n$ = CalculateNodesNumber(*Node*);
**while** $N^n \leq N_i^T$ **do**
  **for** *Node*($j$) in *Node* **do**
    $N_j^p$ = CalculatePointsNumber(*Node*($j$));
    $N_j^H$ = CalculateHighPointsNumber(*Node*($j$));
    **if** $N_j^p > 1$ and $N_j^H \geq 1$ **then**
      $Node^s$ = SplitNode(*Node*($j$));
      Insert($Node^s$, *Node*);
    **else if** $N_j^p = 0$ **then**
      Remove(*Node*($j$), *Node*);
    **else**
      *Node*($j$) $\in$ *Node*;
    **end if**
  **end for**
  $N^n$ = CalculateNodeNumber($Node^n$);
**end while**
$P^R$ = Harris(*Node*);

After the feature extraction process, we use the optimized segmentation maps to reject dynamic feature points. Feature points on dynamic objects are removed directly from the feature points set.

In the rejection thread, ORB-SLAM3 uses a quad-tree algorithm to reject the redundant feature points. However, this method retains a large number of low-quality feature points and reduces the feature matching accuracy of the system. To address this problem, we propose an improved quad-tree algorithm. The algorithm reduces the iterations of low-quality feature points. Firstly, the input layer is divided into 4 nodes. Then, we determine the number of high-quality and low-quality feature points contained in each node, respectively. There are 3 cases as follows:

1) A node is split further into 4 nodes, when the number of feature points in this node is greater than 1 and the number of high-quality feature points in this node is greater than or equal to 1.
2) A node is removed when the number of feature points in the node is equal to 0.
3) A node is no longer divided and is saved, when the feature points in the node are other cases.

Finally, we set the number of feature points required for the $i$-th layer to $N_i^T$. When the total number of nodes is greater than $N_i^T$, all node division operations are finished and one feature point with the largest Harris in each node is retained. Conversely, the above steps are continued. The improved quad-tree algorithm is shown in Algorithm 3. The algorithm takes the $i$-th layer of the pyramid images $L_i$ and the set of high-quality feature points $P^H$ as input. Firstly, the nodes set *Node* is initialized and the number of nodes $N^n$ is calculated by the CalculateNodesNumber function. In the loop body, the number of feature points $N_j^p$ and the number of high-quality feature points $N_j^H$ in *Node*($j$) are calculated respectively. *Node*($j$) is split into 4 new nodes and they is inserted into *Node* if $N^P$ is greater than 1 and $N^H$ is greater than or equal to 1. *Node*($j$) is removed from *Node* if $N^P$ is equal to 0. In other cases, *Node*($j$) remains unchanged and still belongs to *Node*. Finally, only one feature point with the highest Harris value is retained in each node by the Harris function.

The improved quad-tree algorithm slightly reduces the uniformity of feature points, but retains a larger number of high-quality feature points.

## IV. EXPERIMENT AND ANALYSIS

In this section, we experiment the proposed STDC-SLAM on the TUM dataset. Firstly, we introduce the TUM dataset and explain the performance judging criteria of the SLAM system. Then, ablation experiments are conducted to demonstrate the effectiveness of each module. Finally, we compare the performance of STDC-SLAM with other existing state-of-the-art methods on dynamic datasets.

### A. TUM DATASET

The TUM dataset is an excellent dataset for evaluating camera positioning accuracy, and it provides an accurate ground-truth for the sequences.[1] It contains 39 sequences recorded by an RGB-D camera at 30fps with a resolution of 640 × 480. To demonstrate the effectiveness of STDC-SLAM, we use 3 dynamic sequences and 3 static sequences from the TUM dataset to evaluate the performance of STDC-SLAM in a realistic environment, namely w_rpy, w_halfphere, w_xyz, fr1_desk, fr1_desk2, and fr1_room.

[1] https://vision.in.tum.de/data/datasets/rgbd-dataset/download

**TABLE 1.** RMSE(m) of ATE of different systems. OS3 denotes ORB-SLAM3, S denotes STDC network, and Q denotes improved Qtree-ORB algorithm.

| Sequences | OS3 | OS3+S | OS3+S+SRM | OS3+S+SRM+Q (STDC-SLAM) | OS3+Q |
|---|---|---|---|---|---|
| w_rpy | 0.166 | 0.040 | 0.031 | **0.029** | 0.165 |
| w_halfsphere | 0.331 | 0.052 | 0.027 | **0.024** | 0.329 |
| w_xyz | 0.299 | 0.034 | 0.022 | **0.018** | 0.293 |
| fr1_desk | 0.018 | 0.018 | 0.018 | **0.016** | 0.016 |
| fr1_desk2 | 0.025 | 0.025 | 0.025 | **0.023** | 0.023 |
| fr1_room | 0.054 | 0.054 | 0.053 | **0.041** | 0.042 |

### B. JUDGING CRITERIA

To quantitatively evaluate the advantages of the algorithm in this paper, we evaluate the overall performance of the system using Absolute Trajectory Error (ATE). The ATE indicates the global consistency of the trajectory. The Root Mean Square Error (RMSE) reflects the accuracy of the system, while RMSE is computed by:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (X_{g,i} - X_{c,i})^2}{n}} \tag{6}$$

where $n$ means the number of observations, $i$ denotes the $i$-th observation. $X_{g,i}$ is the ground truth of the $i$-th observation, while $X_{c,i}$ is the computation result of the $i$-th observation. Therefore, the RMSE value of ATE is obtained from each sequence to judge the positional accuracy in this paper.

### C. ABLATION EXPERIMENTS

This section describes the ablation experiments that demonstrate the effectiveness of each part of our proposed method.

#### 1) EFFECTIVENESS OF STDC NETWORK COMBINED WITH ORB-SLAM3

We add the STDC network to ORB-SLAM3 as the backbone network of semantic threads. It compares the performance with the original ORB-SLAM3 system in dynamic and static sequences, and the results are shown in columns 2 and 3 of Table 1. In the dynamic sequences w_rpy, w_halfsphere, and w_xyz, the RMSE of ORB-SLAM3 incorporated into STDC (OS3+S) are less than 0.1m. Compared with the original ORB-SLAM3 (OS3), OS3+S is able to operate stably in the dynamic environments. In addition, OS3+S has the same effect as OS3 in the static environment experiments. The results show that ORB-SLAM3 incorporating the STDC network exhibits higher accuracy under dynamic sequences relative to the original ORB-SLAM3 and still shows good stability on static sequences.

Figure 4 shows the experimental results of the original ORB-SLAM3 and the ORB-SLAM3 incorporating the STDC network on the sequence w_xyz from the TUM dataset. It can be seen that the ORB-SLAM3 incorporating the STDC network effectively eliminates the feature points on the human.


**FIGURE 4.** The experimental results of the original ORB-SLAM3 and the ORB-SLAM3 incorporating the STDC network on the sequence w_xyz from the TUM dataset.

#### 2) EFFECTIVENESS OF SRM

The feature points on the dynamic objects are effectively eliminated by using the STDC network. However, there are a small number of dynamic feature points that are not eliminated, as shown in the red box in Figure 5(b). The Segmentation Refinement Module (SRM) is incorporated into our system to further eliminate these remaining dynamic feature points. The experimental results are shown in Figure 5(c).

We experiment ORB-SLAM3 incorporating the semantic segmentation network and SRM under 6 sequences. The results are shown in column 4 of Table 1. In column 4 of Table 1, the RMSE of ORB-SLAM3 incorporating the STDC and SRM modules (OS3+S+SRM) is reduced by 22.5%, 48.1%, and 35.3%, relative to OS3+S in the dynamic sequence, respectively. In addition, OS3+S+SRM still has a stable effect in static sequences. The effectiveness of the SRM module is proved.

#### 3) EFFECTIVENESS OF IMPROVED QTREE-ORB ALGORITHM

To demonstrate the effectiveness of the improved Qtree-ORB algorithm, we compare the original ORB-SLAM3 with the ORB-SLAM3 with the improved Qtree-ORB algorithm on the TUM static sequences fr1_desk, fr1_desk2, and fr1_room. The results are shown in column 6 of Table 1. We find that ORB-SLAM3, which only improves the Qtree-ORB algorithm (OS3+Q), still runs unstable in dynamic sequences. However, in static sequences, the RMSE of OS3+Q is reduced by 11.1%, 8.0%, and 22.2%, compared to OS3, respectively. Also, We separately record the number of high-quality and low-quality feature points as well as the uniformity of feature points distribution in experiments. To quantify the distribution uniformity, the calculation method of literature [35] was used and the results are shown in Table 2. The average number of high-quality and low-quality feature points is 155 and 63, respectively, when extracting feature points per frame using the original ORB-SLAM3. However, when extracting the same number of feature points using the improved ORB-SLAM3, we find that the number of low-quality feature points is significantly reduced, and the average number of low-quality feature points is 35. In addition, the uniformity of feature

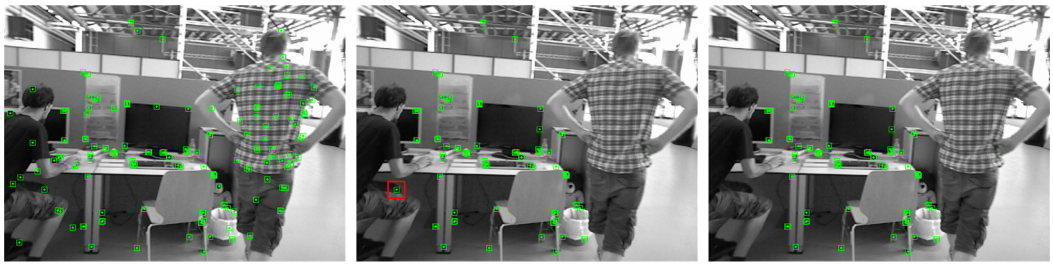


**FIGURE 5.** The experimental results of different ORB-SLAM3 systems on the same frame. (a) ORB-SLAM3. (b) ORB-SLAM3 incorporating the STDC network. (c) ORB-SLAM3 incorporating the STDC network and SRM.

**TABLE 2.** The distribution of feature points of different frames on sequence fr1_desk. H, L respectively denotes the number of high-quality and low-quality feature points, T denotes Total number of feature points, U denotes the uniformity of feature point distribution. U value is larger, the uniformity is worse.

| Frame | ORB-SLAM3 | | | | Improved ORB-SLAM3 | | | |
|---|---|---|---|---|---|---|---|---|
| | H | L | T | U | H | L | T | U |
| 1 | 156 | 63 | 219 | 193 | **164** | **54** | 218 | 201 |
| 60 | 163 | 55 | 218 | 185 | **195** | **23** | 218 | 197 |
| 512 | 171 | 48 | 219 | 180 | **195** | **22** | 217 | 188 |
| 303 | 162 | 55 | 217 | 186 | **185** | **32** | 217 | 194 |
| 482 | 121 | 96 | 217 | 226 | **174** | **44** | 218 | 203 |
| Average | 155 | 63 | 218 | 194 | **183** | **35** | 218 | 203 |

point distribution of the improved algorithm is only slightly reduced compared to the original algorithm. The results show that although the improved Qtree-ORB algorithm slightly reduces the uniformity of feature points distribution, it retains more high-quality feature points and improves the overall quality of feature points.

Finally, we incorporated the STDC network, the SRM, and the improved Qtree-ORB algorithm into ORB-SLAM3, and named it STDC-SLAM. The system was experimented on both dynamic and static datasets, and the results are shown in column 5 of Table 1. The results show that STDC-SLAM improves the location accuracy by more than 82% over ORB-SLAM3 on dynamic sequences and by more than 8% on static sequences.

Figure 6 shows the experimental results of STDC-SLAM on the sequence w_halfphere from the TUM dataset. It can be seen that the system can effectively reject dynamic feature points.

### D. COMPARE WITH STATE-OF-THE-ARTS

Based on the same experimental environment, we compare the ATE and time consumption of STDC-SLAM with DynaSLAM and PSPNet-SLAM. The results are shown in Table 3 and Table 4. We find that the localization accuracy of STDC-SLAM improves on three sequences relative to DynaSLAM, as shown in Table 3. The RMSE of our system in the sequence w_rpy is reduced by 17% compared to DynaSLAM. Although the localization accuracy of STDC-SLAM is not improved in dynamic sequences compared with PSPNet-SLAM, the processing speed of

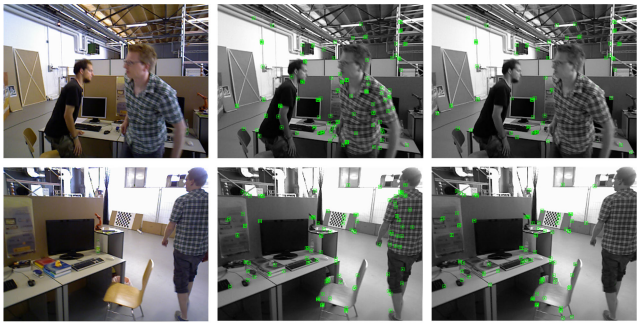


**FIGURE 6.** The experimental results of STDC-SLAM on the sequence w_halfphere from the TUM dataset.

**TABLE 3.** Comparisons of RMSE[m] for our system against the state-of-the-arts in dynamic sequences of TUM RGB-D dataset.

| Sequences | ORB-SLAM3 | DynaSLAM | PSPNet-SLAM | STDC-SLAM |
|---|---|---|---|---|
| w_rpy | 0.166 | 0.035 | 0.026 | 0.029 |
| w_halfsphere | 0.331 | 0.025 | 0.024 | 0.024 |
| w_xyz | 0.299 | 0.018 | 0.017 | 0.018 |

**TABLE 4.** Comparisons of time consumption [ms] for our system against the state-of-the-art in dynamic sequences of TUM RGB-D dataset.

| Sequences | ORB-SLAM3 | DynaSLAM | PSPNet-SLAM | STDC-SLAM |
|---|---|---|---|---|
| w_rpy | 27.96 | 245.14 | 203.13 | 28.43 |
| w_halfsphere | 33.90 | 358.35 | 253.58 | 35.69 |
| w_xyz | 29.16 | 375.89 | 275.36 | 23.74 |

STDC-SLAM is much faster than DynaSLAM and PSPNet-SLAM, as shown in Table 4. The comparison of the estimated trajectories between STDC-SLAM and other SLAM are shown in Figure 7, and STDC-SLAM has higher stability while satisfying the real-time requirements.

## V. CONCLUSION

Based on ORB-SLAM3 and STDC network, we have proposed STDC-SLAM. In the semantic thread, we use STDC network as a semantic segmentation network and add a segmentation refinement module to optimize the segmentation results by using the depth information of the image. This module improves the segmentation accuracy of the system for dynamic objects. In the rejection thread, we have improved the Qtree-ORB algorithm of ORB-SLAM3. The improved

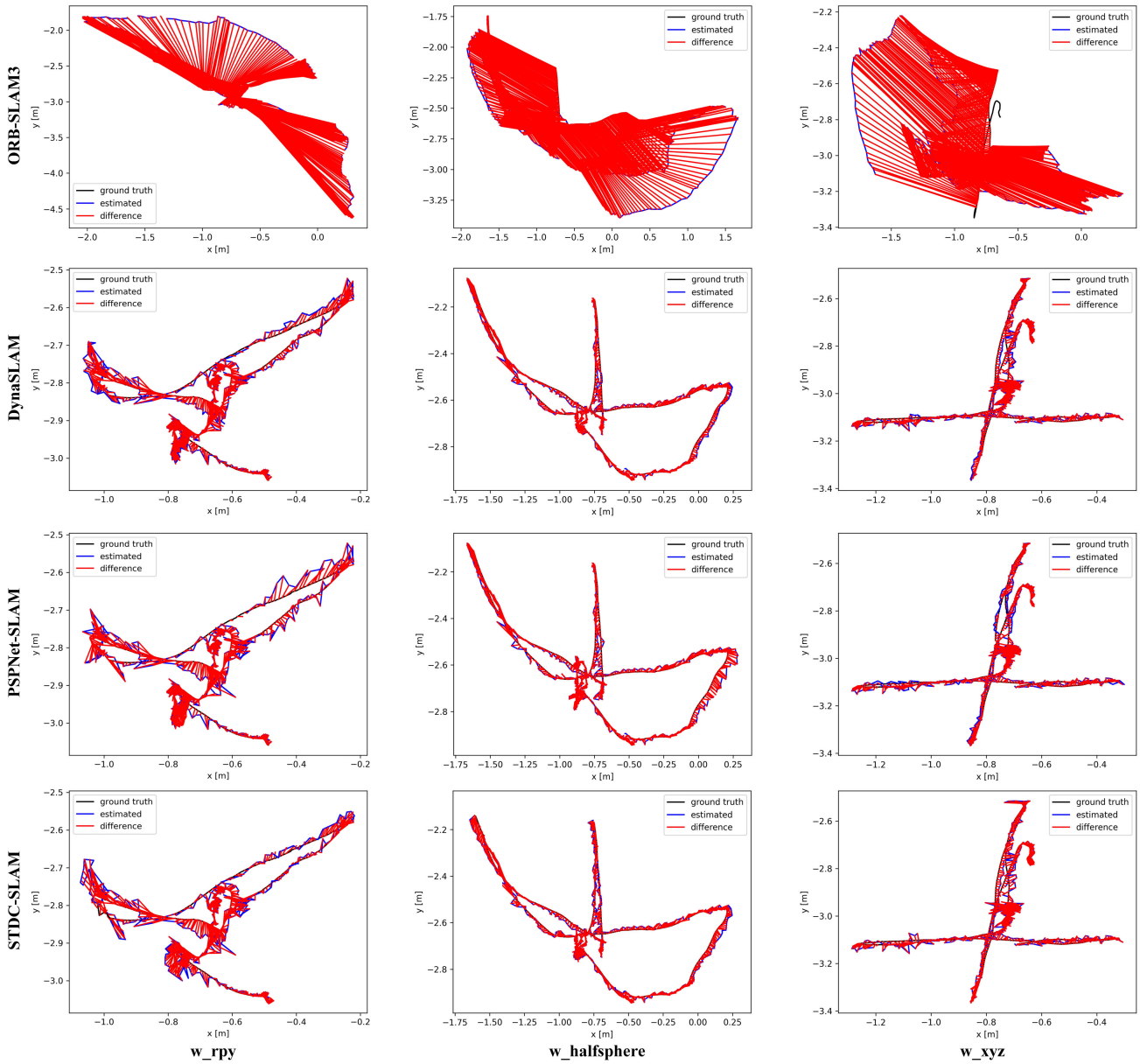


**FIGURE 7.** The comparison of the estimated trajectories between STDC-SLAM and other SLAM.

Qtree-ORB algorithm improves the location accuracy and robustness of the system. Finally, the effectiveness of each module is verified on the TUM dataset, and the absolute trajectory error and time consumption are compared with other excellent SLAM systems. The results show that our proposed system has higher localization accuracy and satisfies the real-time performance.

Although our proposed system has made some progress in accuracy and real-time, there are still some tasks we need to do. On the one hand, the segmentation accuracy of the semantic segmentation network needs to be further improved. On the other hand, our proposed system easily leads to tracking failure in a highly dynamic environment. In the future, we need to improve the accuracy of the semantic segmentation network, and continue to experiment STDC-SLAM on different datasets to improve the robustness of the system in dynamic environments.

## REFERENCES

[1] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 2, no. 3, pp. 194–220, Sep. 2017.

[2] R. Li, S. Wang, and D. Gu, "Ongoing evolution of visual SLAM from geometry to deep learning: Challenges and opportunities," *Cognit. Comput.*, vol. 10, no. 6, pp. 875–889, Dec. 2018.

[3] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[4] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[5] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.
[6] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis.*, vol. 8690. Cham, Switzerland: Springer, Sep. 2014, pp. 834–849.
[7] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.
[8] Y. Sun, M. Liu, and M. Q.-H. Meng, "Improving RGB-D SLAM in dynamic environments: A motion removal approach," *Robot. Auton. Syst.*, vol. 89, pp. 110–122, Mar. 2017.
[9] 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, Dec. 2016.
[10] X. Long, W. Zhang, and B. Zhao, "PSPNet-SLAM: A semantic SLAM detect dynamic object by pyramid scene parsing network," *IEEE Access*, vol. 8, pp. 214685–214695, 2020.
[11] Y. Fan, Q. Zhang, Y. Tang, S. Liu, and H. Han, "Blitz-SLAM: A semantic SLAM in dynamic environments," *Pattern Recognit.*, vol. 121, Jan. 2022, Art. no. 108225.
[12] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "DS-SLAM: A semantic visual SLAM towards dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1168–1174.
[13] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder–decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Jan. 2017.
[14] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2881–2890.
[15] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "BiseNet: Bilateral segmentation network for real-time semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 325–341.
[16] M. Fan, S. Lai, J. Huang, X. Wei, Z. Chai, J. Luo, and X. Wei, "Rethinking BiSeNet for real-time semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9716–9725.
[17] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
[18] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, Nov. 2007, pp. 225–234.
[19] A. Kundu, K. M. Krishna, and J. Sivaswamy, "Moving object detection by multi-view geometric techniques from a single camera mounted robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 4306–4312.
[20] R. Wang, W. Wan, Y. Wang, and K. Di, "A new RGB-D SLAM method with moving object detection for dynamic indoor scenes," *Remote Sens.*, vol. 11, no. 10, p. 1143, 2019.
[21] Y. Fan, H. Han, Y. Tang, and T. Zhi, "Dynamic objects elimination in SLAM based on image fusion," *Pattern Recogn. Lett.*, vol. 127, pp. 191–201, Nov. 2018.
[22] Y. Fang and B. Dai, "An improved moving target detecting and tracking based on optical flow technique and Kalman filter," in *Proc. 4th Int. Conf. Comput. Sci. Educ.*, Jul. 2009, pp. 1197–1202.
[23] M. C. Bakkay, M. Arafa, and E. Zagrouba, "Dense 3D SLAM in dynamic scenes using Kinect," *Pattern Recognit. Image Anal.*, vol. 9117, pp. 121–129, Jan. 2015.
[24] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6243–6252.
[25] X. Li and R. Belaroussi, "Semi-dense 3D semantic mapping from monocular SLAM," 2016, *arXiv:1611.04144*.
[26] B. Bescos, J. M. Fácil, J. Civera, and J. L. Neira, "DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 4076–4083, Oct. 2018.
[27] Y. Ai, T. Rui, M. Lu, L. Fu, S. Liu, and S. Wang, "DDL-SLAM: A robust RGB-D SLAM in dynamic environments combined with deep learning," *IEEE Access*, vol. 8, pp. 162335–162342, 2020.
[28] Q. Jin, Z. Meng, T. D. Pham, Q. Chen, L. Wei, and R. Su, "DUNet: A deformable network for retinal vessel segmentation," 2018, *arXiv:1811.01206*.
[29] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Feb. 2004.
[30] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. 9th Eur. Conf. Comput. Vis.*, vol. 3951, May 2006, pp. 404–417.
[31] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 105–119, Jan. 2010.
[32] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Proc. Eur. Conf. Comput. Vis.*, 2010, vol. 63, no. 14, pp. 778–792.
[33] D. Sun, S. Zhang, and Y. Wang, "Improved feature point extraction and mismatch eliminating algorithm," *Syst. Sci. Control Eng.*, vol. 8, no. 1, pp. 11–21, Jan. 2020.
[34] G. Li, L. Yu, and S. Fei, "A deep-learning real-time visual SLAM system based on multi-task feature extraction network and self-supervised feature points," *Measurement*, vol. 168, pp. 108403–108412, Jan. 2021.
[35] J. Yao, P. Zhang, and C. Luo, "ORB feature uniform distribution algorithm based on improved quadtree," *Comput. Eng. Des.*, vol. 41, no. 6, pp. 1629–1634, 2020.

**ZHANGFANG HU** received the master's degree from the University of Electronic Science and Technology, Sichuan, China, in 1994. She was a Visiting Scholar at Zhejiang University, China. She is currently a Professor with the College of Photoelectrics, Chongqing University of Posts and Telecommunications (CQUPT). Her research interests include photoelectric sensing and optoelectronic information processing.

**JIAN CHEN** received the B.S. degree from the Chongqing University of Posts and Telecommunications (CQUPT), Chongqing, China, in 2020, where he is currently pursuing the master's degree with the College of Photoelectrics. His current research interests include mobile robots and semantic SLAM.

**YUAN LUO** received the M.S. degree from the Chongqing University of Posts and Telecommunications (CQUPT), Chongqing, China, in 1996, and the Ph.D. degree from Chongqing University, Chongqing, in 2003. She was a Visiting Scholar at the Université de Montréal, Canada, in 2006. She is currently a Professor with CQUPT. Her research interests include computer vision, photoelectric sensing, image processing, and mobile robots.

**YI ZHANG** received the M.S. degree from the Hefei University of Technology (HFUT), Anhui, China, in 1997, and the Ph.D. degree from the Huazhong University of Science and Technology (HUST), Hubei, in 2002. He was a Visiting Scholar at the University of Essex, U.K., from 2004 to 2005. He is currently a Professor with the Chongqing University of Posts and Telecommunications (CQUPT). His research interests include intelligent systems, mobile robots, and intelligent logistics technology and equipment.