IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 10, 2020, accepted June 15, 2020, date of publication June 17, 2020, date of current version June 26, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3003160

Semantic SLAM With More Accurate Point Cloud Map in Dynamic Environments

YINGCHUN FAN[1,2], QICHI ZHANG[1], SHAOFENG LIU[1], YULIANG TANG[1], XIN JING[2], JINTAO YAO[2], AND HONG HAN[1,2], (Member, IEEE)
[1]School of Artificial Intelligence, Xidian University, Xi'an 710071, China
[2]Shaanxi Key Laboratory of Integrated and Intelligent Navigation, Xi'an 710071, China

Corresponding author: Hong Han (hanh@mail.xidian.edu.cn)

This work was supported by the open fund of Shaanxi Key Laboratory of Integrated and Intelligent Navigation under Grant SKLIIN-20180102 and Grant SKLIIN-20180107.

ABSTRACT Static environment is a prerequisite for most existing vision-based SLAM (simultaneous localization and mapping) systems to work properly, which greatly limits the use of SLAM in real-world environments. The quality of the global point cloud map constructed by the SLAM system in a dynamic environment is related to the camera pose estimation and the removal of noise blocks in the local point cloud maps. Most dynamic SLAM systems mainly improve the accuracy of camera localization, but rarely study on noise blocks removal. In this paper, we proposed a novel semantic SLAM system with a more accurate point cloud map in dynamic environments. We obtained the masks and bounding boxes of the dynamic objects in the images by BlitzNet. The mask of a dynamic object was extended by analyzing the depth statistical information of the mask in the bounding box. The islands generated by the residual information of dynamic objects were removed by a morphological operation after geometric segmentation. With the bounding boxes, the images can be quickly divided into environment regions and dynamic regions, so the depth-stable matching points in the environment regions are used to construct epipolar constraints to locate the static matching points in the dynamic regions. In order to verify the preference of our proposed SLAM system, we conduct the experiments on the TUM RGB-D datasets. Compared with the state-of-the-art dynamic SLAM systems, the global point cloud map constructed by our system is the best.

INDEX TERMS Dynamic environment, global point cloud map, noise blocks, semantic SLAM.

I. INTRODUCTION

Simultaneous localization and Mapping (SLAM) plays an important role in the field of autonomous robots and unmanned vehicles [1]. The main purpose of SLAM is to use sensors such as cameras to construct the environment model without prior knowledge of the scene and to estimate the pose and motion trajectory of the carrier itself [2]. Recent years have seen the great development of visual SLAM, which can be classified into feature-based indirect SLAM systems [3]–[5] and direct ones based on photometric error [6], [7].

A common assumption for most of these visual SLAM systems is that the environment is static, which greatly limits the practical application of these systems. There are many dynamic objects in the real environment, such as pedestrians, vehicles, etc. When these dynamic objects enter the camera's field of view, the pose estimation of the camera is directly interfered [8], and information of these dynamic objects will be preserved in the constructed map [9]. It's impossible to use this kind of contaminated map for robot navigation or human-computer interaction.

The main idea of these SLAM systems work in the dynamic environments is to classify the static and dynamic points in the environment, using the static points to estimate the pose of the camera and construct the environment map [10]. The dynamic points can be directly discarded or tracked according to the requirements of the task.

With the development of deep learning, great progresses have been made in target recognition and segmentation, so some researchers have combined the semantic information of the objects in the environment with SLAM [11]–[15]. The main idea of the current semantic SLAM systems working in the dynamic environment is to define the potential dynamic objects in advance, obtaining semantic information of the

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

objects in images through a deep CNN, and then combine the semantic with geometric information to segment dynamic regions accurately. Finally, the systems use the static parts of the environment to estimate the camera pose.

In this paper, we proposed a novel semantic SLAM system with more accurate point cloud map in dynamic environments. The semantic information of our proposed SLAM system is provided by BlitzNet [16], which can simultaneously generate bounding boxes and masks of the potential dynamic objects, such as people. In order to remove the noise blocks formed by the leaked information of the dynamic objects, we extend the masks of the dynamic objects by the depth statistical information of the masks in the bounding boxes of the dynamic objects. Then a geometric segmentation is operated on the depth image. The residual information of the dynamic objects which is still not included in the extended masks can be segmented as some islands, and we can delete these islands by a simple morphological operation to obtain the clean local point map. In the feature matching stage, we only use depth-stable feature points, which can effectively eliminate the influence of missing depth values and a sudden change of depth values. The images are divided into environment regions and dynamic ones by the bounding boxes of the dynamic objects. The static matching points in the dynamic regions can be located by the epipolar constraint constructed by inliers in the environment regions. The main contributions of this paper are as follows:

1) We extend the mask of the dynamic object to include more information about the object.
2) A bidirectional search strategy is proposed to track the bounding box of the dynamic object.
3) We integrate our approach with ORB-SLAM2 system [5]. Evaluations and method comparisons are performed with the TUM RGB-D dataset [17]. Our SLAM system can obtain clean and accurate global point cloud maps in both highly and lowly dynamic environments.

The rest of the paper is organized as follows: Section II discusses the related work. Section III presents our proposed method. Section IV shows the experiment results and discussion. Section V, finally, concludes the work.

II. RELATED WORK

A. DYNAMIC SLAM

CoSLAM [18] divides points into four types: static, dynamic, false and uncertain based on the assumption that the projected position of a static point in space follows a Gaussian distribution, and the type of the point can be changed according to the motion state of the object. At the same time, the system adopts a multi-camera combination observation strategy to effectively deal with the situation that the number of the static points captured by a certain camera is small or does not exist at all. Evers and Naylor [19] proposed a GEM-SLAM based on probability hypothesis density filters for dynamic scenes, the method probabilistically anchors the observer state by fusing the observer information inferred from the scene with the observer motion reports. Bahraini *et al.* [20] proposed an approach to segment and track multiple dynamic objects based on the multilevel-RANSAC algorithm. DMS-SLAM [21] uses the sliding window model to achieve feature matching between two discontinuous image frames, and adopts the Grid-based Motion Statistics (GMS) algorithm [22] to filter the initial feature matching points. This approach not only eliminates the impact of the dynamic objects, but also has more feature matching points than ORB-SLAM2, which has a great advantage for estimating camera pose accurately.

RGB-D camera can provide both vision and depth information of the scene and is convenient to install. In recent years, many researchers have used RGB-D camera to study SLAM in dynamic environment. Sun *et al.* [23] proposed a motion removal approach based on RGB-D data as a preprocessing stage of DVO SLAM [24] to filter out data related to moving objects. The approach consists of two on-line parallel processes: learning and inference. The functions of these two processes are to construct and update the foreground model, and perform pixel-wisely segmentations on the foreground model. Scona *et al.* [25] segmented the input RGB-D image pair into K geometric clusters by applying K-means on the 3D coordinates of the scene points, and assume that each geometric cluster behaves as a rigid body, the segmentation of static and dynamic objects are converted to the analysis of the states of geometric clusters. This method focuses on building the static environment model rather than analyzing the motion state of objects. Li and Lee [26] proposed a SLAM system based on the static point weight. Static point weight indicates the possibility that a point is part of the static environment. Kim and Kim [27] proposed a nonparametric background model from depth scenes, which can reduce influence of dynamic objects, and the motion of camera is estimated by an energy-based dense-visual-odometry approach based on the background model. Dai *et al.* [28] used the consistency of point's geometric correlations to resist the interference caused by moving objects, and the geometric correlations between map points are created by Delaunary triangulation. The dynamic objects can be separated from static environment by removing non-consistency connections.

In order to deal with the motion blur caused by high-speed motion of camera in highly dynamic environments, the event camera was introduced into the research of SLAM by the researchers at the University of Zurich [29]–[33], and the output of the event camera is an asynchronous stream of events. However, because the output of the event camera is different from the traditional camera output, the traditional visual algorithm cannot be directly applied to the SLAM system built on the event camera, and the higher price also limits the use of the sensor.

B. SEMANTIC SLAM

Dyna-SLAM [34], MaskFusion [35], MID-Fusion [36] and a RGB-D SLAM system proposed by Zhao *et al.* [37] use Mask R-CNN [38] as the semantic segmentation approach.

Dyna-SLAM removes all the potential dynamic objects such as people, cars, and animals. Considering that some dynamic objects cannot be detected by Mask R-CNN, because they are not previously defined as potential dynamic objects, such as a rotated chair, or a book hold by a moving person, the authors utilize multi-view geometry to locate these movable objects. The working environment of MaskFusion is mainly indoors, so the authors proposed two strategies to judge whether an object is dynamic or static: firstly, consistence of object motion; secondly, object in contact with people is movable. MID-Fusion consists of four parts: segmentation, tracking, fusion and raycasting. The system creates sub-maps for every possibly rigidly moving object in the environment, and fuses the geometric, semantic, and motion properties information of dynamic objects into these sub-maps. And in the process of camera tracking, MID-Fusion discards the matching points in the human mask area. The system proposed by Zhao *et al.* refined the boundaries of the detected dynamic objects by integrating the Canny edges of the RGB images with the mask boundaries, because the contours of the dynamic objects obtained by the semantic segmentation are not precise.

DS-SLAM [39], SOF-SLAM [40] and SDF-SLAM [41] use SegNet [42] as the semantic segmentation algorithm. The authors of DS-SLAM assume that the feature points on the people are most likely to be outliers, so they exclude the people in the images and construct epipolar line constraint by the matching points in the environment regions. Then the constraint is utilized to detect whether the people are static. If a person is determined to be static, then matching points on the person can be used to predict the pose of the camera. SOF-SLAM proposed a dynamic features detection algorithm which utilizes the semantic information to aid the calculation of epipolar geometry, and this system can remove the dynamic feature points effectively. SDF-SLAM is the continuation of SOF-SLAM, which outperforms SOF-SLAM, because SDF-SLAM solves these two problems: first, the matching points on the slow-moving objects may be recognized as static matching points, and second, the dynamic information in adjacent frames is easy to be interfered by noise. Brasch *et al.* [43] proposed a joint probabilistic model based on the semantic prior information provided by a CNN, using temporal motion information to determine the state of a certain map point. This method can deal with the slow-moving and temporarily static objects effectively. Sun *et al.* [44] proposed a movable object aware SLAM system via weakly supervised semantic segmentation, and the main advantage of this system is that it avoids expensive annotations for training. DDL-SLAM [45] detects dynamic objects with semantic masks obtained by DUNet [46] and multi-view geometry, and then reconstructs the background obscured by dynamic objects with the strategy of image inpainting.

Instead of obtaining masks of potential dynamic objects, some researchers directly utilize the bounding boxes to remove the dynamic regions. Yang *et al.* [47] used Faster-RCNN [48] to detect the potential dynamic objects, and refined the data association by removing the mismatching points, so the camera pose are calculated by the better data association and the graph optimization. Zhang *et al.* [49] used YOLO [50] to detect and recognize objects, so that the relationship between keyframes and objects are built to filter the dynamic feature points and generate semantic maps. Dynamic-SLAM [51] uses an SSD [52] object detector to detect the dynamic objects defined by a dynamic characteristic score based on life experience, and the system adopts a compensation algorithm based on the speed invariance in adjacent frames to deal with missing detection. Aiming at the three problems that the dynamic SLAM needs to solve, namely obtaining accurate camera localization, getting navigation maps and real-time segmentation of dynamic objects, Sun *et al.* [53] proposed a multi-purpose dynamic SLAM framework which is compatible with different segmentation methods for different purposes and situations.

Summarizing the above SLAM systems working in dynamic environment, it can be found that most of these works focus on improving the camera positioning accuracy, but there is almost no research on noise blocks removal in the obtained point cloud map. The quality of the global map constructed by SLAM is related to two factors. One is the accuracy of camera pose estimation, and the other is whether the noise blocks in the map are effectively removed. Since the existing semantic segmentation algorithms are not perfect [35], some information of the dynamic objects will leak into the environment, and this information will be retained in the constructed local point cloud maps to form a large number of noise blocks. These noise blocks will greatly affect the actual use of the map. In addition, if these point cloud maps with a large number of noise blocks are converted to other forms of maps, such as octomaps [54], the quality of maps will not be significantly improved.

III. SYSTEM DESCRIPTION

The overview of our proposed semantic SLAM system in dynamic environments is shown in Fig. 1. First of all, the RGB images pass through a CNN (Convolution Neural Network) that performs object detection and pixel-wise segmentation at the same time. Considering the practical application of the system, the selection of CNN for semantic segmentation of potential dynamic objects requires a balance between real-time and accuracy. We choose BlitzNet, which is a real-time deep neural network that performs object detection and semantic segmentation in one-time forward propagation, as the basic network of our semantic SLAM system. The detected objects include people, tvmonitors, chairs, etc., which are common indoors. We roughly divide the objects in the environment into three categories: dynamic objects, such as people; potential dynamic objects, such as chairs, books, keyboards, whose status are determined by the dynamic objects; static objects, such as tvmonitors, whose positions in the environment are relatively fixed and do not change easily.

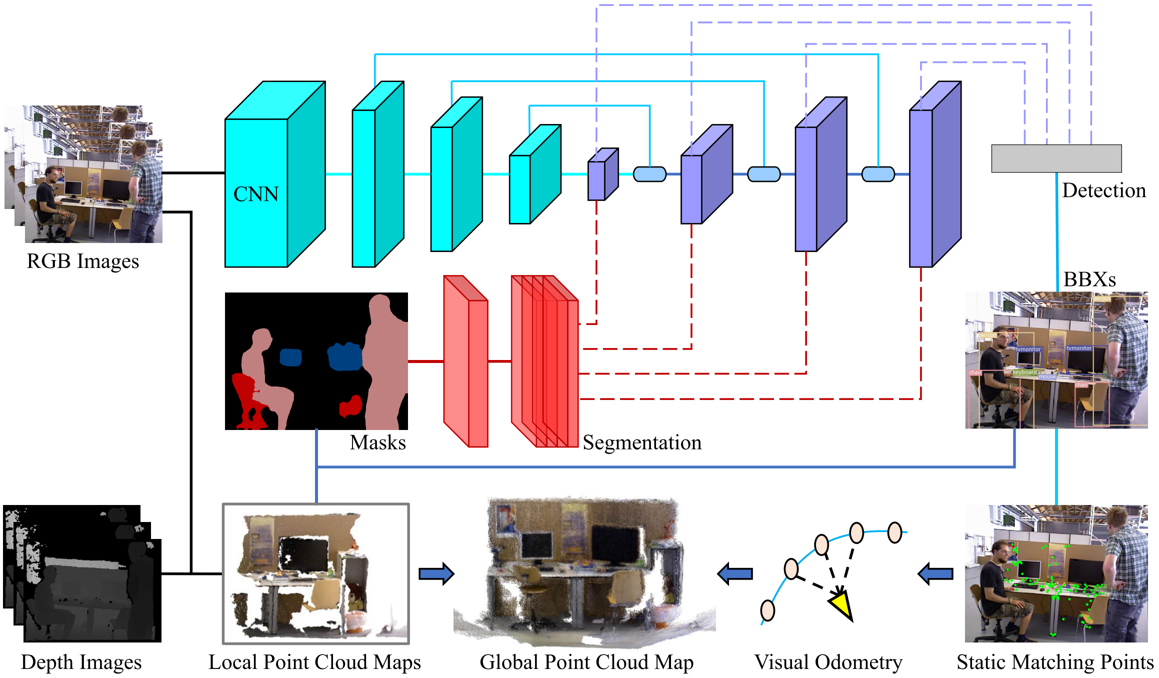


FIGURE 1. Overview of the system.

The essence of the global point cloud map P_G constructed by SLAM system is to stitch the local point cloud P_L^i obtained by each group of key frames, namely

$$P_G = \sum_{i=1}^{n} \left(R_i P_L^i + t_i\right) \tag{1}$$

where, n is the total number of key frames, $i = 1, \ldots, n$, R_i and t_i are the rotation matrix and translation matrix for converting the local point cloud map to the coordinate system where the global point cloud map is located. The values of R_i and t_i are determined by the camera's pose in space.

The quality of the global cloud map obtained after stitching in a dynamic environment is related to two factors: first, the accuracy of camera pose estimation; second, the removal of noise blocks formed by the dynamic objects in the local point cloud map.

Smears will occur in the constructed point cloud map because of the dynamic objects. Although removing the image information corresponding to the mask regions of dynamic objects can improve this situation, the dynamic object mask obtained by the existing semantic segmentation is not perfect. Especially the edges of the dynamic object might not be included in the mask, so these parts would leak into the environment. The noise blocks formed by this edge information greatly affect the quality of the point cloud map, and even cause the point cloud map to look chaotic. In this paper, we extend the mask of dynamic object through analyzing the depth information of the mask within the bounding box of the dynamic object. The extended mask can contain as much information of the dynamic object as possible. Then we segment the depth image by geometric method, and remove the islands formed by the residual information of the dynamic objects. After that, a clean local point cloud map can be obtained.

In a dynamic environment, the main reason for the large error of camera pose estimation is that the matching points on the dynamic objects participate in the camera positioning process. Especially when the dynamic objects occupy a large space in the image and the texture information on the dynamic objects is rich, the matching points cannot be effectively removed by traditional algorithms such as RANSAC [55]. In this paper, we quickly divide the image into the environment regions and dynamic regions by the bounding boxes of the dynamic objects, and the matching points in the environment regions are used to construct epipolar constraint to locate the static matching points in the dynamic regions, so as to ensure that the final matching points for camera localization are static.

A. DYNAMIC OBJECT MASK EXTENSION

The masks of the dynamic objects obtained by BlitzNet are not complete, and some information of the dynamic objects will leak into the environment. In this section we take the human mask extension as an example. As shown in Fig. 2, when comparing the RGB image with the obtained human mask image, it can be clearly seen that parts of the sitting person's body are not included in the mask, which are marked by the red boxes in the figure. In order to observe the information of the human bodies leaking into the environment more intuitively, we set the depth values of the areas in the depth

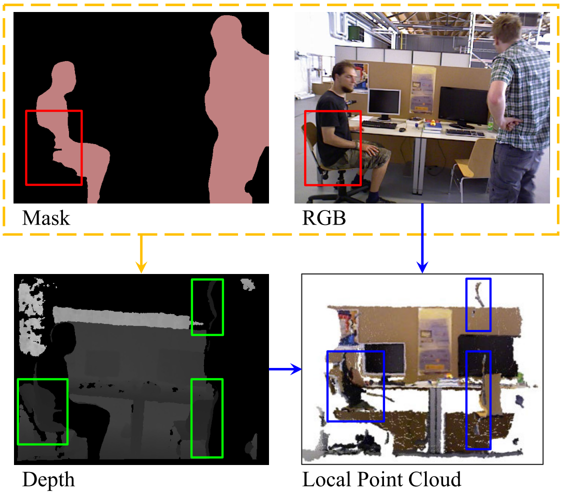


FIGURE 2. Human information leaked into the environment.

image corresponding to the human masks to 0. And the parts of the human bodies leaked into the environment are marked with green boxes. It can be seen from the depth image that a part of the edge of the walking person is also leaked into the environment. A lot of noise blocks will exist in the local point cloud generated by such depth image and RGB image, which are marked with blue boxes.

We use the depth statistical information in the human mask regions to find the pixels that belong to the human body but leak into the environment within the human's bounding box, that is, to extend the human body mask. As shown in Fig. 3, the depth values corresponding to the two human mask regions are counted. Considering that there may be some pixels with a depth value of 0 and some noise in the human mask regions, the 0 value is not counted in the process of statistics. After that, the outliers would be removed. $D_{P(i)}$ is used to represent the set of depth values within the human mask, where $P(i)$ represents the person's label that appears in the image. The maximum and minimum values in $D_{P(i)}$ are used as the upper and lower bounds of the depth value in the bounding box of $P(i)$ respectively, that is

$$U_d = \max\left(D_{P(i)}\right) \tag{2}$$

$$L_d = \min\left(D_{P(i)}\right) \tag{3}$$

Human mask can be extended according to (4)

$$\begin{cases} L_d \le D_{P(i)}^{BBX}(u,v) \le U_d, & (u,v) \in M_{P(i)} \\ else, & (u,v) \notin M_{P(i)} \end{cases} \tag{4}$$

where $D_{P(i)}^{BBX}(u,v)$ denotes the depth value of the pixel (u,v) in the bounding box of $P(i)$, $M_{P(i)}$ represnts the mask of $P(i)$.

If a dynamic object is far away from the camera, or the segmentation algorithm performs poorly on a certain type of object, the area of the obtained mask would be very small, so the depth values in the mask would not be enough to represent the depth information of the dynamic object effectively. In this case, we remove the information in the whole bounding box of the dynamic object to eliminate its impact on the local point cloud map, as follows:

$$\begin{cases} S_{DO(i)}^{mask} < \tau_1, & S_{DO(i)}^{mask} = S_{DO(i)}^{BBX} \\ else, & S_{DO(i)}^{mask} = S_{DO(i)}^{mask} \end{cases} \tag{5}$$

where $S_{DO(i)}^{mask}$ represents the mask area of the i-th dynamic object, $S_{DO(i)}^{BBX}$ denotes the area of the bounding box of the i-th dynamic object, i is the number of the detected dynamic object, τ_1 is a preset threshold value, in this paper $\tau_1 = 5000$, the unit is pixel.

At the same time, we have noticed that some contents of the dynamic object may leak into the environment beyond the bounding box. The shape of this leaked information is usually long and narrow. Due to the continuous movement of the dynamic object in the environment, the difference between depth values in the edge of the dynamic object in the two adjacent frames is much larger than that of the environment. We can use this feature to remove these narrow and long edges, as follows:

Subtract the previous depth image from the current depth image and take the absolute value, as shown below:

$$d\_sub = |F_C - F_P| \tag{6}$$

If the value of $d\_sub(u,v)$ is large, there are two situations for pixel (u,v). One is the point at the edge of the object in the environment. This type of point can be removed because there will be a lot of redundant information in the process of constructing the global point cloud map, and the depth value at the edge of an object changes a lot, which is the reason for the obvious layering at the object edge in the obtained point cloud. Another is the point that the dynamic object leaked into the environment. In this paper, the current frame is operated as follows:

$$\begin{cases} d\_sub(u,v) > \tau_2, & F_C(u,v) = 0 \\ else, & F_C(u,v) = F_C(u,v) \end{cases} \tag{7}$$

where τ_2 is a preset threshold value, in this paper $\tau_2 = 5000$. The dynamic object mask extension algorithm processing is shown in Algorithm 1.

B. INTERACTION JUDGMENT BETWEEN POTENTIAL DYNAMIC OBJECT AND DYNAMIC OBJECT

The state of the potential dynamic object is determined by whether the dynamic object interacts with it. For example, when someone adjusts the position of a chair or sits on it, we should regard the chair at this time as a dynamic object. If the chair is isolated in the environment, it can be considered to be a static object. When a chair is a dynamic object, the camera cannot use the matching points on it for pose estimation, and the information of it should be removed from the constructed point cloud map. When a chair is a static object, the matching points detected on it can participate in the camera pose estimation, and the information of it is also an essential part of the point cloud map. So, it is necessary to judge the status of the potential dynamic object.

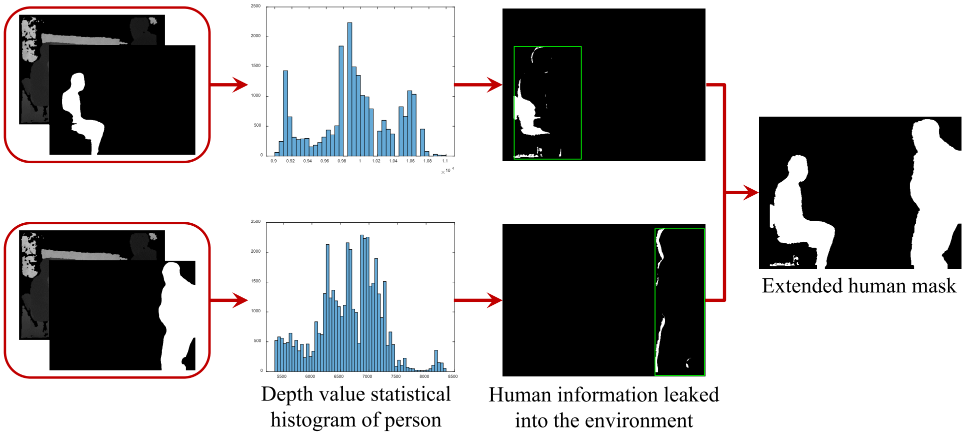

Extended human mask
Depth value statistical histogram of person
Human information leaked into the environment

FIGURE 3. Human mask extension.

Algorithm 1 Dynamic Object Mask Extension Algorithm

```
Input: Current depth image $F_C$, previous depth image $F_P$,
    bounding box of the $i$-th dynamic object $BBX_{DO(i)}$, mask
    of the $i$-th dynamic object $M_{DO(i)}$;
Output: Processed mask $M'_{DO(i)}$, processed depth image
    $F'_C$;
$D_{DO(i)} = RecordDepthValue\left(F_C, M_{DO(i)}\right)$;
Remove outliers in $D_{DO(i)}$;
$L_d = GetMin\left(D_{DO(i)}\right)$, $U_d = GetMax\left(D_{DO(i)}\right)$;
for each point $(u, v)$ within $BBX_{DO(i)}$ do
    if $L_d \le GetDepth\,(u, v) \le U_d$ then
        $M'_{DO(i)} \leftarrow (u, v)$;
    end if
end for
$S^{mask}_{DO(i)} = CalculateArea\left(M_{DO(i)}\right)$;
if $S^{mask}_{DO(i)} \le \tau_1$ then
    $M'_{DO(i)} = GetArea\left(BBX_{DO(i)}\right)$;
end if
$F'_C = F_C$;
$d\_sub = |F_C - F_P|$;
for each point $(u, v)$ within $d\_sub$ do
    if $d\_sub\,(u, v) > \tau_2$ then
        $F'_C\,(u, v) = 0$;
    end if
end for
```

People are the main dynamic objects in the indoor environment, so we should consider that the state of a potential dynamic object to be dynamic, when it is in contact with a person. We utilize the bounding boxes and the depth information in the masks of the person and the potential dynamic object to judge the status of the potential dynamic object. The flow diagram is shown in Fig. 4.

First, we label people and potential dynamic objects in the image and get the corresponding labels, such as $\{P\,(1)\,, \ldots, P\,(n)\}$, $\{O\,(1)\,, \ldots, O\,(k)\}$.

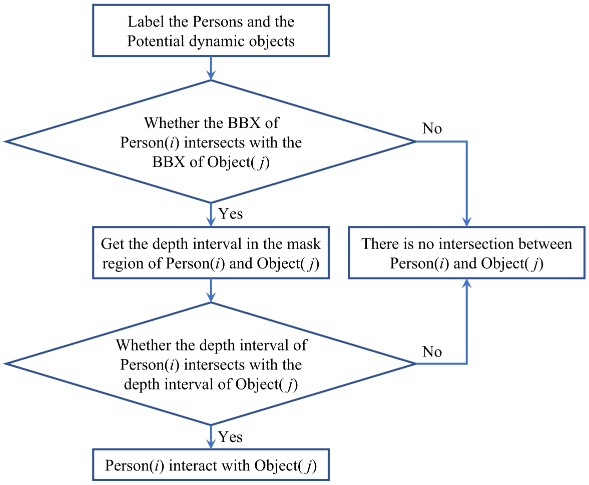

Label the Persons and the Potential dynamic objects
Whether the BBX of Person(*i*) intersects with the BBX of Object(*j*)
No
Yes
Get the depth interval in the mask region of Person(*i*) and Object(*j*)
There is no intersection between Person(*i*) and Object(*j*)
Whether the depth interval of Person(*i*) intersects with the depth interval of Object(*j*)
No
Yes
Person(*i*) interact with Object(*j*)

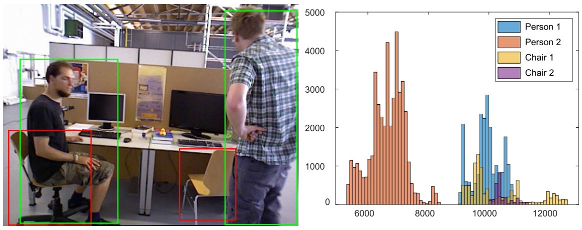
FIGURE 4. Flow diagram of interaction judgment between human and potential dynamic object.

When the bounding box of $P\,(i)$ intersects that of $O\,(j)$, it is considered that $P\,(i)$ may interact with $O\,(j)$, and the label group $\{P\,(i)\,, O\,(j)\}$ is saved. Otherwise, the label group $\{P\,(i)\,, O\,(j)\}$ is deleted, as follows:

$$\begin{cases} S^{BBX}_{P(i)} \cap S^{BBX}_{O(j)} \neq \emptyset, & Save\,\{P\,(i)\,, O\,(j)\} \\ else, & Delete\,\{P\,(i)\,, O\,(j)\} \end{cases} \tag{8}$$

where $S^{BBX}_{P(i)}$, $S^{BBX}_{O(j)}$ respectively represent the areas occupied by the bounding boxes of $P\,(i)$ and $O\,(j)$ in the image, $i = 1, \ldots, n, j = 1, \ldots, k$.

Next, for the label group $\{P\,(i)\,, O\,(j)\}$ saved in the previous step, the depth information is used to further determine the intersection between $P\,(i)$ and $O\,(j)$. We count the depth values in the masks of $P\,(i)$ and $O\,(j)$, and obtain the sets of depth values $D^{Mask}_{P(i)}$ and $D^{Mask}_{O(j)}$ in the masks of $P\,(i)$ and $O\,(j)$ respectively after removing the 0 values and the outliers. The label groups $\{P\,(i)\,, O\,(j)\}$ saved by (9) are those which



(a) (b)

FIGURE 5. Judgment of interaction between human and chair. (a) Bounding boxes of people and chairs; (b) Histogram of depth values within the masks of the people and the chairs.

intersect with each other.

$$\begin{cases} \max\left(D_{O(j)}^{Mask}\right) < \min\left(D_{P(i)}^{Mask}\right), & Delete\left\{P\left(i\right), O\left(j\right)\right\} \\ \max\left(D_{P(i)}^{Mask}\right) < \min\left(D_{O(j)}^{Mask}\right), & Delete\left\{P\left(i\right), O\left(j\right)\right\} \\ else, & Save\left\{P\left(i\right), O\left(j\right)\right\} \end{cases} \quad (9)$$

Taking the case shown in Fig. 5 as an example, the dynamic objects in Fig. 5(a) are two persons which are marked by green boxes, while a swivel chair and an ordinary chair are the potential dynamic objects which are marked by red boxes. The labels of the sitting person and the walking one are $P\left(1\right)$ and $P\left(2\right)$, respectively. $C\left(1\right)$ and $C\left(2\right)$ represent the swivel chair and the ordinary one. It is obvious in Fig. 5(a) that the bounding box of the sitting person intersects with that of the swivel chair, and the bounding box of the walking person intersects with that of the ordinary chair. That is, the label groups need to be saved are $\left\{P\left(1\right), C\left(1\right)\right\}$ and $\left\{P\left(2\right), C\left(2\right)\right\}$.

Next, the depth values within the masks of the two people and the two chairs are counted respectively. For the convenience of observation, we display the statistical histograms of the depth values of the two people and two chairs on one graph, as shown in Fig. 5(b). It can be seen from Fig. 5(b) that the depth interval of $P\left(1\right)$ intersects with that of $C\left(1\right)$ and $C\left(2\right)$. According to the label groups saved in the previous step, we can know that the label group that needs to be actually saved is $\left\{P\left(1\right), C\left(1\right)\right\}$, that is, $P\left(1\right)$ interacts with $C\left(1\right)$.

C. BOUNDING BOX TRACKING

In some cases, Blitz-Net cannot effectively detect the dynamic object, because the dynamic object is too small, or because only part of the dynamic object appear in the image, as shown in Fig. 6. Therefore, it is necessary to re-detect the dynamic objects when the detection is missing, that is, to track the bounding boxes of the dynamic objects.

Based on the assumption that the moving speed of the dynamic object relative to the camera is constant for a short period of time, this paper proposes a bidirectional search strategy to track the bounding box of the dynamic object.

$D_{indicator}^{K_t} = \{0, 1\}$ is used as the indicator of whether Blitz-Net detects dynamic object in the K_t frame. When $D_{indicator}^{K_t} = 0$, it indicates that no dynamic object is detected in the frame. If $D_{indicator}^{K_t} = 1$, it denotes that the dynamic object D_{target} is detected in the K_t frame. Let $BBX^{K_t} = \left(x_{tlc}^{K_t}, y_{tlc}^{K_t}, x_{lrc}^{K_t}, y_{lrc}^{K_t}\right)$ represent the bounding box of dynamic object D_{target} in the K_t frame, where $\left(x_{tlc}^{K_t}, y_{tlc}^{K_t}\right)$ are the upper left corner coordinates of the bounding box, and $\left(x_{lrc}^{K_t}, y_{lrc}^{K_t}\right)$ are the lower right corner ones.

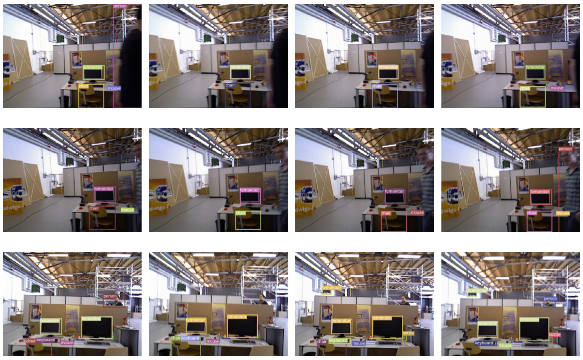


FIGURE 6. Examples of Blitz-Net unable to effectively detect dynamic objects.

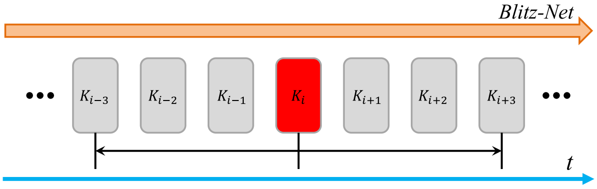


FIGURE 7. Bounding box tracking for dynamic object.

As shown in Fig. 7, the red rectangle represents the current frame K_i. When no dynamic object is detected in K_i, that is $D_{indicator}^{K_i} = 0$, then 3 frames backward and 3 frames forward are searched. The bounding box of the dynamic object in the current frame K_i can be tracked by the values recorded in the indicators of the 6 frames. How to determine whether there is a dynamic object in the current frame, and if there is a dynamic object, how to get the bounding box of the object, as follows:

1) Dynamic object is detected in the previous 3 frames and the later 3 frames, the dynamic object is considered to exist in the current frame K_i, and the bounding box of the dynamic object can be obtained by

$$BBX^{K_i} = BBX^{K_{i-np}} + \frac{\left(BBX^{K_{i+nl}} - BBX^{K_{i-np}}\right) * np}{(nl + np)} \quad (10)$$

where K_{i-np} and K_{i+nl} are the previous frame and the later frame closest to K_i, and a same dynamic object is detected in K_{i-np} and K_{i+nl}, $np \leq 3$, $nl \leq 3$.

2) When the dynamic object is detected only in the previous 3 frames or only in the later 3 frames, the dynamic object is considered to exist in the current frame K_i, and the bounding box of the dynamic object in the image closest to K_i is used as the bounding box in K_i.

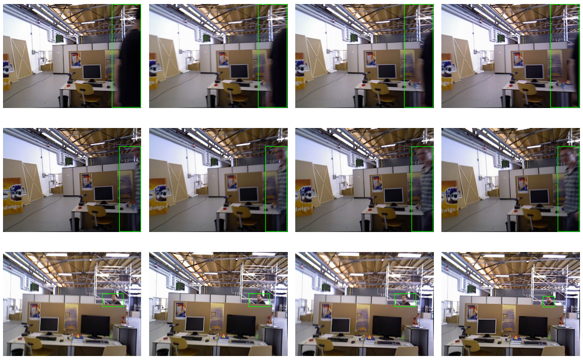


FIGURE 8. Tracking results of dynamic object bounding box.

3) When Blitz-Net does not detect dynamic object in these 6 frames, we consider that there is no dynamic object in the current frame K_i.

The proposed bounding box tracking algorithm can effectively track the bounding box of the dynamic object missed by Blitz-Net in Fig. 6. The results are shown in Fig. 8.

D. GEOMETRIC SEGMENTATION OF DEPTH IMAGE

After removing the dynamic objects, we found that some of the information of these objects is still left in the environment. In order to remove this residual information, geometric segmentation is performed on the depth image. The residual information is usually some small isolated patches in the segmented depth image, which can be removed by a simple morphological operation. In the depth image, the depth of the junction between different objects is not continuous, that is, the depth value between the objects and the background changes a lot. According to this property, the segmentation edges of the depth image can be placed in the depth discontinuities.

Our segmentation method for depth image is as follows:

We traverse the depth image with a slider of size 2 ∗ 2. Image coordinates corresponding to the pixel in the upper left corner of the slider is (u, v), and depth values in the slider are recorded, as follows:

$$D_b = d\,(u : u + 1, v : v + 1) \tag{11}$$

where d represents the depth image. The depth image can be segmented quickly by the following formula:

$$\begin{cases} \max(D_b) - \min(D_b) > \tau_3, & d\,(u, v) = 0 \\ else, & d\,(u, v) = d\,(u, v) \end{cases} \tag{12}$$

where τ_3 is a preset threshold value, in this paper $\tau_3 = 500$.

Perform area statistics on the image patches with depth information in the segmented depth image to obtain $S_{patch(i)}$, $i = 1, \ldots, m$, where m is the number of image patches. The islands formed by the residual information of the dynamic objects can be removed as follows:

$$\begin{cases} S_{patch(i)} \leq \tau_4, & Delete\,\{patch\,(i)\} \\ else, & Save\,\{patch\,(i)\} \end{cases} \tag{13}$$

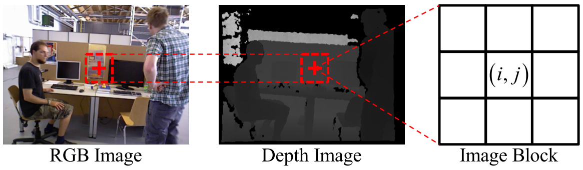


FIGURE 9. Image block centered on the integer pixel coordinates of the feature point.

where τ_4 is a preset threshold value, in this paper $\tau_4 = 1000$, the unit is pixel.

E. FEATURE POINTS WITH STABLE DEPTH VALUES

Assume that we get two sets of matching points $A = \{P_{a1}, \ldots, P_{an}\}$ and $B = \{P_{b1}, \ldots, P_{bn}\}$. The external parameter matrix of the camera can be obtained by solving the least squares problem shown below:

$$\min_{R,t} \sum_{i=1}^{N} \|P_{ai} - (RP_{bi} + t)\|^2 \tag{14}$$

For two adjacent frames of depth images, there are regions where the depth values are missing, and the depth values of these regions are 0. Matching points in these regions cannot provide any useful information for solving ICP (Iterative Closest Point). In addition, there is a sudden change in the depth values of feature points at the edge of objects, which will directly affect the solution of (14). At the same time, the depth values of some matching points on the dynamic targets will also change greatly. In this paper, we solve (14) by using matching points with stable depth values.

Feature points with stable depth values are usually on the surface of certain objects, such as the desk baffle region marked by the red dotted frame [56], as shown in Fig. 9. The red cross on the RGB image represents the location of a detected feature point, an image block of size 3 ∗ 3 centered on (i, j) is taken on the depth image for later process, (i, j) are the integer pixel coordinates of the feature point.

Firstly, we detect whether the depth value in the image block corresponding to each feature point is missing. If there is a pixel with a depth value of 0 in the image block, the corresponding feature point is considered to be in the region where the depth value is missing, and the feature point is deleted. Fig. 10 is a detailed view of part regions where feature points with missing depth values in Fig. 9. It can be seen from the figure that some parts in the computer keyboard region lose depth values. Although the roof region contains a lot of texture information, the depth values of the feature points in this region are completely missing, since it is far from the camera and exceeds the effective measurement range of the depth camera.

Next, we calculate the standard deviation of the depth values in the image block corresponding to each feature point that is retained. The standard deviation of the depth values is

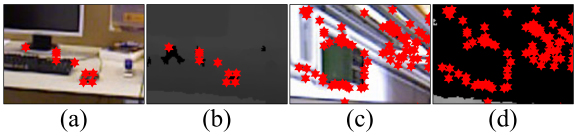


FIGURE 10. Feature points with missing depth values. (a) Feature points detected in the computer keyboard region. (b) Depth value of the computer keyboard region is partially missing. (c) Feature points detected in the roof region. (d) Depth value of the roof region is completely missing.

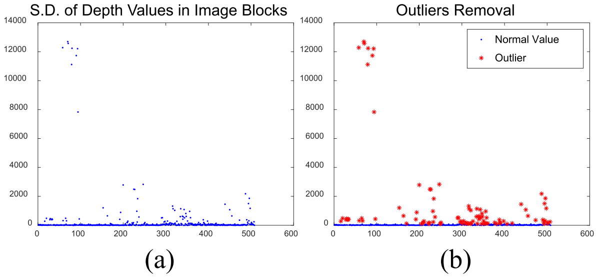


FIGURE 11. Find image blocks with stable depth values. (a) Standard deviations of depth values in image blocks. (b) Eliminate outliers in the sequence.

typically small in the image block where the depth values are stable, while large where there is a sudden change in depth values. Feature points corresponding to the image blocks with large standard deviation of depth values can be deleted by setting an appropriate threshold.

In the course of experiments, we found that the number of image blocks with sudden change in depth values is usually much less than that with stable depth values. All standard deviations obtained are stored in a sequence, and the feature points corresponding to the outliers in the sequence are eliminated. An outlier value is defined as a value that is more than three scaled MAD (median absolute deviations) away from the median. As shown in Fig. 11, the red asterisk represents the outlier that need to be rejected.

Fig. 12 is a detailed view of part regions where feature points with sudden changes in depth values in Fig. 9. These regions with sudden changes in depth values are mainly distributed at the edges of objects. And the depth values on the surfaces of objects at a distance obtained by the depth camera will also be inaccurate. In fact, the accuracy of the depth value of the object obtained by the depth camera is inversely proportional to the distance between the target and the camera. The closer the distance, the more accurate the depth value. In the process of constructing the point cloud map, we discard points whose depth value exceeding 30000 (6m).

After the above two steps, the feature points with stable depth values can be obtained. As shown in Fig. 13, the three types of feature points obtained are displayed on the RGB and depth images respectively, the green point represents the feature point with stable depth value, the red point denotes the feature point with missing depth value, and the blue point indicates the feature point with sudden change in depth value.

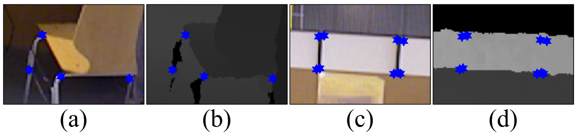


FIGURE 12. Feature points with sudden changes in depth values. (a) Feature points detected at the edge of a chair. (b) There is a sudden change in the depth values at the edge of the chair. (c) Feature points detected on a plank at a longer distance. (d) Inaccurate measurement of depth values on the plank.

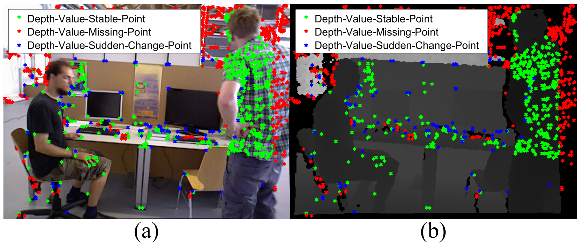

Depth-Value-Stable-Point
Depth-Value-Missing-Point
Depth-Value-Sudden-Change-Point
Depth-Value-Stable-Point
Depth-Value-Missing-Point
Depth-Value-Sudden-Change-Point
(a)
(b)

FIGURE 13. Three types of feature points. (a) Feature points on the RGB image. (b) Feature points on the depth image.

When matching feature points between a pair of images, we only use those with stable depth values. It can be seen from Fig. 13 that there are a large number of green points on the walking person, which is the main reason for the large error of the camera pose estimation.

F. LOCATION OF STATIC MATCHING POINTS

After obtaining the dynamic objects, we can use the bounding boxes of the dynamic objects to segment the image quickly, and divide the image into dynamic regions and environment regions. The feature points in the image can be divided into 4 groups after feature matching: inliers set P_E^I in the environment regions, outliers set P_E^O in the environment regions, dynamic points set P_D^D in the dynamic regions, static points set P_D^S in the dynamic regions.

After matching the feature points in the environment regions of the previous frame and the current frame, P_E^O can be effectively removed by RANSAC algorithm, and the fundamental matrix F between the two adjacent frames can be calculated by P_E^I. By matching the feature points in the dynamic regions of the previous frame and the current frame, the matching points $P_D^P = \left[u_D^P, v_D^P, 1\right]$, $P_D^C = \left[u_D^C, v_D^C, 1\right]$ can be got. The distance from the matching points to the corresponding epipolar line can be calculated by

$$d = \frac{\left|P_D^C F \left(P_D^P\right)^T\right|}{\sqrt{l_x^2 + l_y^2}} \tag{15}$$

where l_x and l_y can be got by

$$\left[l_x, l_y, l_z\right]^T = F\left(P_D^P\right)^T \tag{16}$$

Each group of matching points in the dynamic regions can get a distance d_i, of which i is the serial number of the group of matching points, and the set to which the i-th group

TABLE 1. Results of absolute trajectory error (ATE).

Sequences	ORB-SLAM2		Dyna-SLAM		DS-SLAM		Ours	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
fr3/w/half	0.4543	0.2524	0.0296	0.0157	0.0303	0.0159	**0.0241**	**0.0122**
fr3/w/rpy	0.5391	0.2283	**0.0354**	**0.0190**	0.4442	0.2350	0.0453	0.0316
fr3/w/static	0.3194	0.1819	**0.0068**	**0.0032**	0.0081	0.0033	0.0077	0.0039
fr3/w/xyz	0.7521	0.4712	0.0164	0.0086	0.0247	0.0161	**0.0157**	**0.0083**
fr3/s/static	0.0087	0.0043	0.0108	0.0056	**0.0065**	**0.0033**	0.0080	0.0037

TABLE 2. Results of translational relative pose error (RPE).

Sequences	ORB-SLAM2		Dyna-SLAM		DS-SLAM		Ours	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
fr3/w/half	0.3216	0.2629	0.0284	0.0149	0.0297	0.0152	**0.0274**	**0.0140**
fr3/w/rpy	0.3880	0.2823	**0.0448**	**0.0262**	0.1503	0.1168	0.0616	0.0357
fr3/w/static	0.1928	0.1773	**0.0089**	0.0044	0.0102	**0.0038**	0.0102	0.0049
fr3/w/xyz	0.4834	0.3663	0.0217	0.0119	0.0333	0.0229	**0.0204**	**0.0107**
fr3/s/static	0.0095	0.0046	0.0126	0.0067	**0.0078**	**0.0038**	0.0087	**0.0038**

TABLE 3. Results of rotational relative pose error (RPE).

Sequences	ORB-SLAM2		Dyna-SLAM		DS-SLAM		Ours	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
fr3/w/half	6.6515	5.3990	0.7842	0.4012	0.8142	0.4101	**0.7440**	**0.3459**
fr3/w/rpy	7.5906	5.4768	**0.9894**	**0.5701**	3.0042	2.3065	1.3831	0.8319
fr3/w/static	3.5991	3.2457	**0.2612**	0.1259	0.2690	0.1215	0.2631	**0.1119**
fr3/w/xyz	8.8419	6.6762	0.6284	0.3848	0.8266	0.2826	**0.6227**	**0.3807**
fr3/s/static	0.2881	0.1244	0.3416	0.1642	**0.2735**	0.1215	0.2782	**0.1210**

of matching points belongs can be judged according to the following formula:

$$\begin{cases} d_i > \tau_5, & i \in P_D^D \\ else, & i \in P_D^S \end{cases} \tag{17}$$

where τ_5 is a preset threshold value, in this paper $\tau_5 = 0.5$. The matching points groups belong to P_D^D are directly discarded, and those belong to P_D^S but are not on the dynamic object mask can participate in camera pose estimation.

IV. EXPERIMENTAL RESULTS

Our system adopts ORB-SLAM2 [5], which is one of the most outstanding SLAM systems based on the feature points matching, as the global SLAM solution. Dyna-SLAM [34] and DS-SLAM [39], the two best solutions for SLAM in highly dynamic environments are both built on ORB-SLAM2.

In this section, we will compare the proposed system with ORB-SLAM2, Dyna-SLAM and DS-SLAM on the five sets of sequences selected from TUM RGB-D dataset. These five sets of sequences include four sets of walking sequences, mainly for our experiments, and a set of sitting sequences, which are selected as the reference group.

In the walking sequences, two persons walk back and forth in the scene, occasionally sitting on the chairs talking and gesturing, so they can be regarded as highly dynamic objects. The walking sequences divided into four groups according to the different movement modes of the camera, which are halfsphere, rpy, static and xyz. Halfsphere means that the camera motion following a halfsphere-like trajectory; rpy means that the camera rotated along the roll-pitch-yaw axes; static means that the camera roughly kept in place manually; xyz means that the camera moved along the x-y-z axes. For the convenience of expression, we use fr3/w/half, fr3/w/rpy, fr3/w/static and fr3/w/xyz to represent the four sets of walking sequences. In the sitting sequences, the two persons just moved only a little bit relative to the environment, most of the time sitting on chairs chatting and gesturing. In this paper, we choose fr3/s/static as the reference group.

A. EVALUATION OF THE CAMERA LOCATION

Metrics Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) are used for quantitative comparison, and the experimental results are shown in Table 1 – Table 3. The values of Root Mean Square Error (RMSE) and Standard Deviation (S.D.) are presented in the tables; RMSE measures the deviation between the observed value and the true value

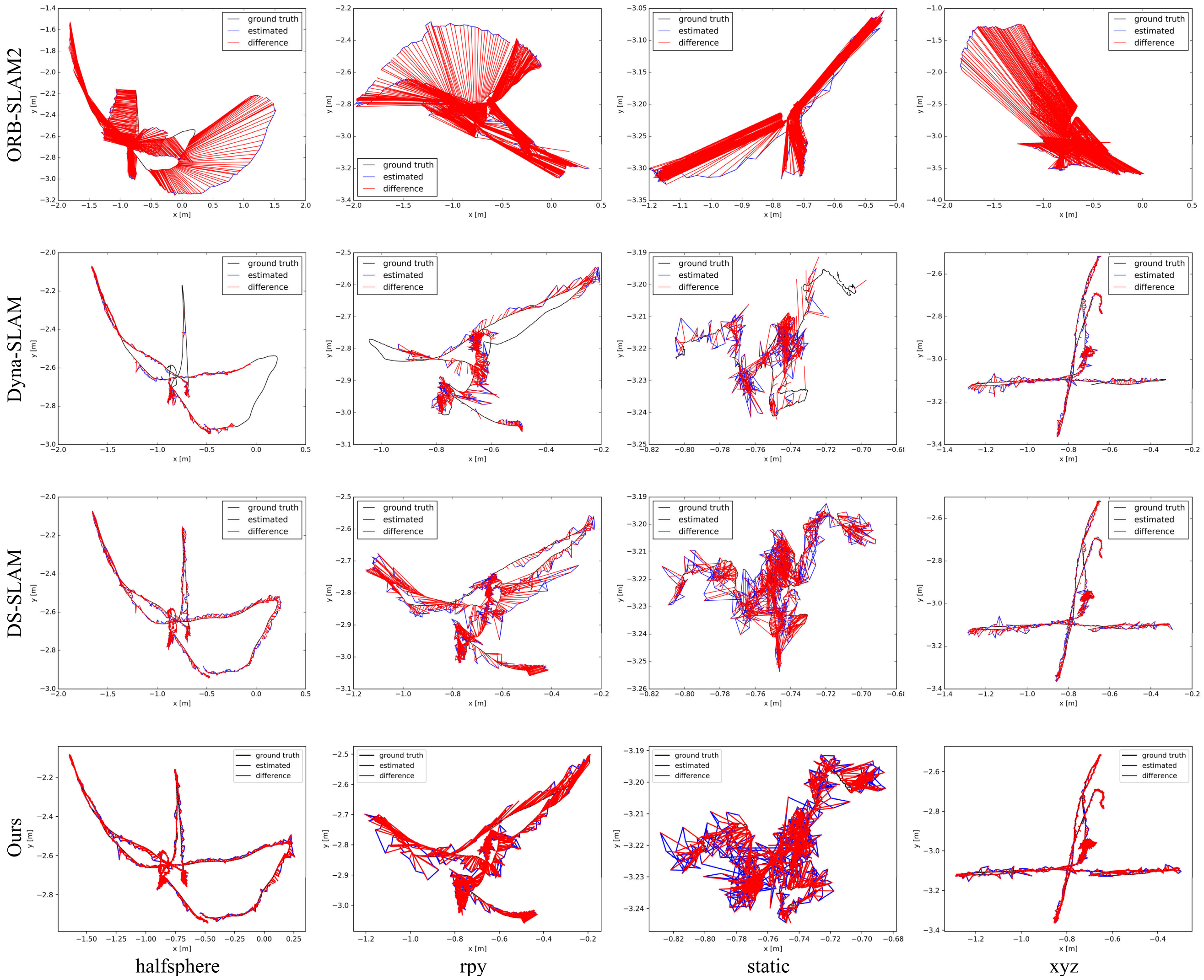


FIGURE 14. Comparison of the estimated trajectories.

and S.D. reflects the extent of deviation for a group as a whole. The two values indicate the robustness and stability of SLAM systems, respectively.

Table 1 gives the results of Absolute Trajectory Error (ATE). ORB-SLAM2 cannot handle with the highly dynamic scenes effectively, and the other three systems have greatly improved compared with ORB-SLAM2. Our proposed system achieved the best results on fr3/w/half and fr3/w/xyz, and the results obtained on fr3/w/rpy and fr3/w/static are close to the results of Dyna-SLAM.

Table 2 presents the results of Translational Relative Pose Error (RPE). On fr3/w/half and fr3/w/xyz, the results of our system are the best. On fr3/w/static, the results of our system, DS-SLAM and Dyna-SLAM are very close. The RMSE value of Dyna-SLAM and the S.D. value of DS-SLAM achieved the best results respectively, and the RMSE value of DS-SLAM is the same as our system.

Table 3 provides the results of Rotational Relative Pose Error (RPE). Our system got the best results on fr3/w/half and fr3/w/xyz. Dyna-SLAM achieved the best results on fr3/w/rpy, but our system is better than DS-SLAM and ORB-SLAM2. It should be noticed that the RMSE values of the three dynamic SLAM systems on fr3/w/rpy were not obvious improvement. On fr3/w/static, the RMSE value of Dyna-SLAM and S.D. value of our system got the best results respectively, and the RMSE value of our system is better than DS-SLAM. In fact, the results of the three systems are very close to each other. The RMSE value of our system and the S.D. value of DS-SLAM achieved the best results on fr3/w/xyz respectively. According to the results on the fr3/w/rpy, it can be inferred that the performance on the rotation angle estimation of SLAM systems is greatly challenged in a highly dynamic environment when the camera motion mode is rotating along the roll-pitch-yaw axes.

TABLE 4. Results of successfully tracked trajectory points.

Sequences	Total	ORB-SLAM2		Dyna-SLAM		DS-SLAM		Ours	
		Tracked	Ratio	Tracked	Ratio	Tracked	Ratio	Tracked	Ratio
fr3/w/half	1021	942	99.3%	525	51.4%	**1018**	**99.7%**	**1018**	**99.7%**
fr3/w/rpy	866	825	95.3%	546	63.1%	**864**	**99.8%**	**864**	**99.8%**
fr3/w/static	717	**714**	**99.6%**	375	52.3%	**714**	**99.6%**	**714**	**99.6%**
fr3/w/xyz	827	809	97.8%	757	91.5%	**826**	**99.9%**	**826**	**99.9%**
fr3/s/static	679	675	99.4%	675	99.4%	**676**	**99.6%**	**676**	**99.6%**

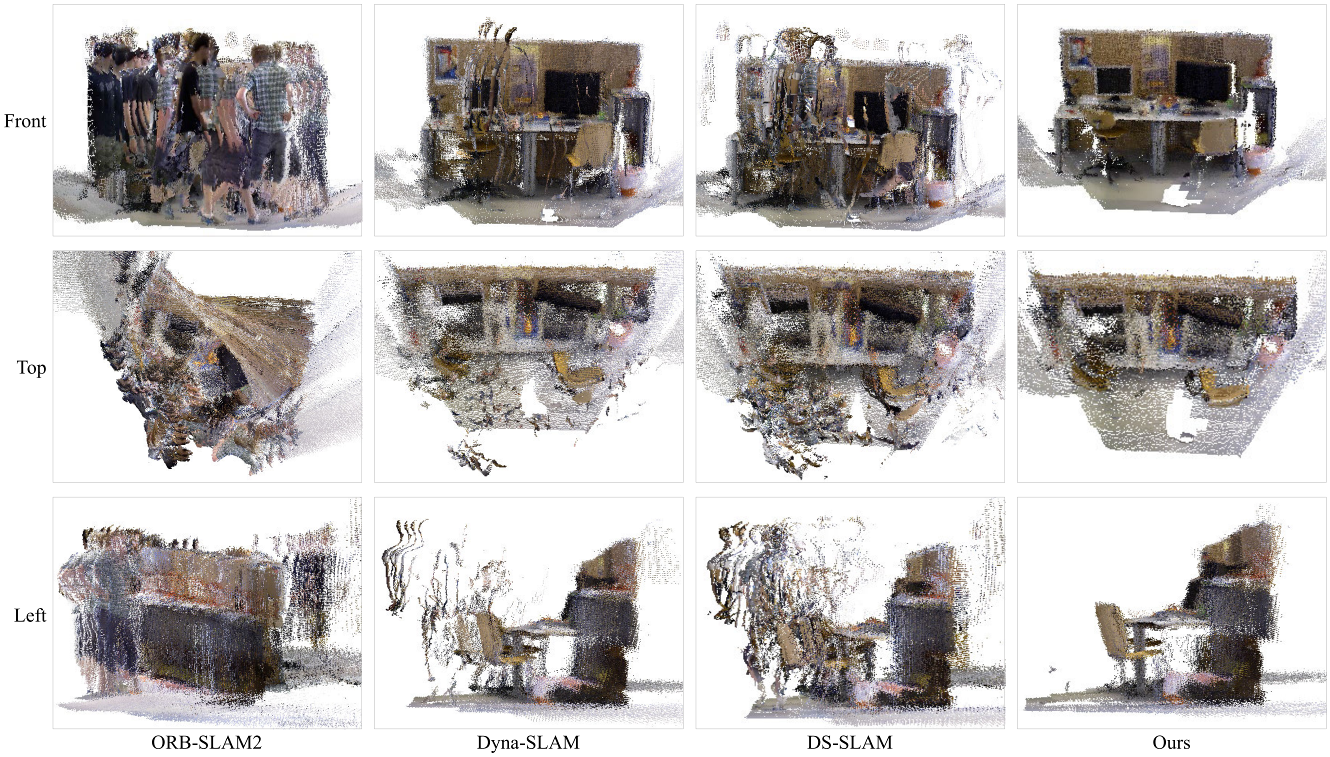


FIGURE 15. Comparison of global point cloud maps constructed by the four SLAM systems on fr3/w/xyz.

As can be seen from Table 1 – Table 3, the results of the three dynamic SLAM systems on fr3/s/static are not much different from ORB-SLAM2, so we conclude that the ORB-SLAM2 can handle the camera location problem in lowly dynamic environment well.

Fig. 14. shows the estimated trajectories of ORB-SLAM2, Dyna-SLAM, DS-SLAM and our system compared with the ground-truth. As can be seen from the first row images, in highly dynamic environments, the trajectories generated by ORB-SLAM2 have large errors compared with the real trajectories. Dyna-SLAM, DS-SLAM and our system have achieved good results compared with ORB-SLAM2. On fr3/w/half, fr3/w/rpy and fr3/w/static, the trajectories generated by Dyna-SLAM are not complete compared with the other three SLAM systems, as shown in the second row.

Table 4 gives the results of successfully tracked trajectory points of the four SLAM systems. As we can see from Table 4, our system tracked the same number of trajectory points on the five sequences as that tracked in DS-SLAM.

B. EVALUATION OF THE GLOBAL POINT CLOUD MAP

First, we show the global point cloud maps constructed by the four SLAM systems in a highly dynamic environment. Taking the global point cloud map obtained on fr3/w/xyz as an example, as shown in Fig. 15.

From the front view of the global point cloud map obtained by ORB-SLAM2, we can see that the information of the two persons are remained in the global point cloud map, and other objects in the environment such as the table, tvmonitors and chairs are obscured by these smears. It can be seen from the top view that the plank of the table is twisted, the reason for this phenomena is that the pose estimation of the camera has a large error, causing the points on the plank to be mapped to the incorrect position when constructing the point cloud map.

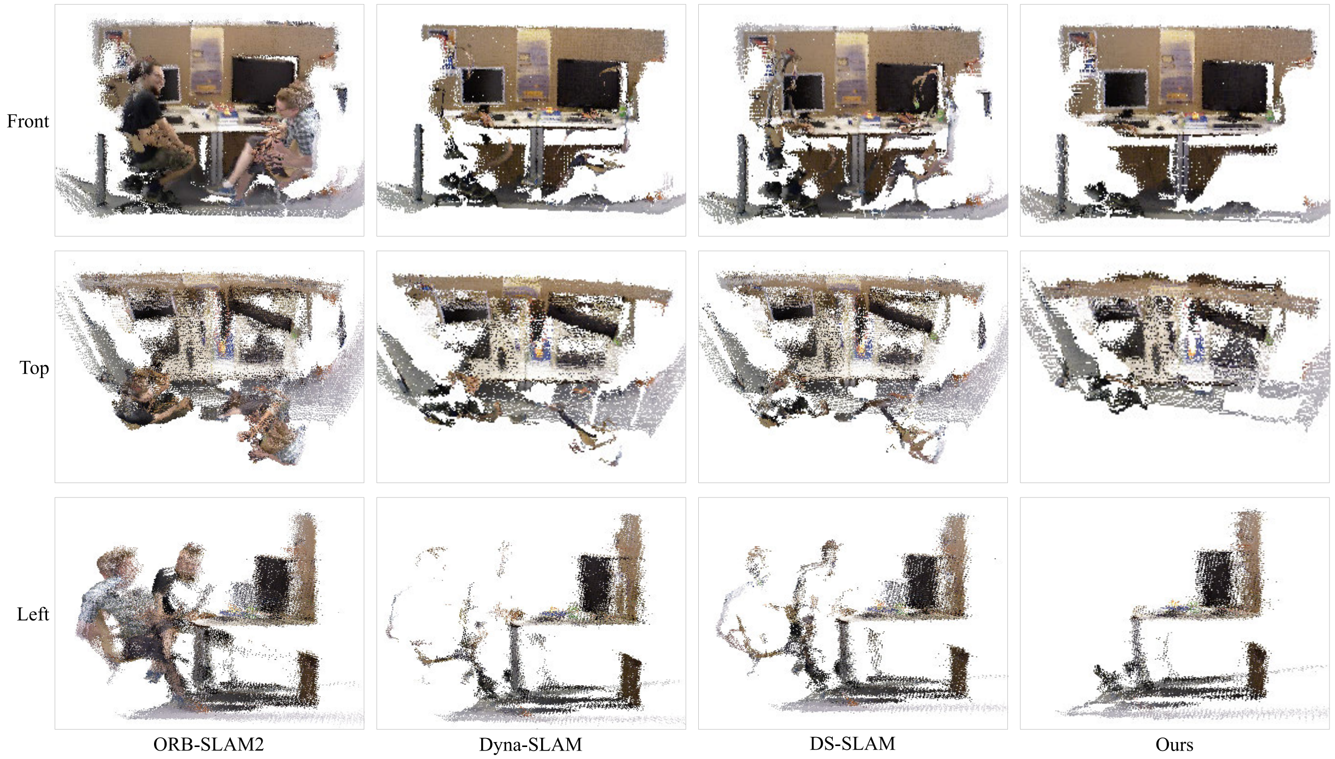


FIGURE 16. Comparison of global point cloud maps constructed by the four SLAM systems on fr3/s/static.

In fact, the map looks so chaotic, and it is impossible to use this map for robot navigation or human-computer interaction.

The camera pose estimation accuracy is greatly improved after removing the interference of the dynamic objects, so compared with ORB-SLAM2, the quality of the global point cloud maps constructed by Dyna-SLAM, DS-SLAM and our system is greatly improved. In the global point cloud maps constructed by these three dynamic SLAM systems, we can clearly see the chairs, screens and other targets in the environment.

However, as can be seen from the images in the second and third columns, due to the lack of operations to remove noise blocks, the information leaked into the environment by these two people exists in the global point cloud maps obtained by Dyna-SLAM and DS-SLAM. The amount of noise blocks is directly related to the masks obtained through the dynamic objects segmentation algorithm used by the two dynamic SLAM systems. The more information contained in the dynamic object mask, the less information the dynamic object leaks into the environment. The noise blocks in the global point cloud map of Dyna-SLAM are mainly some slender edges, while that of DS-SLAM are coarser.

As can be seen from the fourth columns of images, after the operations of removing the noise blocks, the information leaked by these two people into the environment has been effectively removed in our global point cloud map, that is, our SLAM system can construct a clean and accurate global point cloud map in a highly dynamic environment.

Then we show the global point cloud maps constructed by the four SLAM systems in a lowly dynamic environment. Taking the global point cloud map obtained on fr3/s/static as an example, as shown in Fig. 16.

As can be seen from the first column of images, although the global point cloud map of ORB-SLAM2 retains the information of these two people, we can clearly see the objects in the environment, and there is no distortion in the plank of the table. As the conclusion in section IV-A, the camera pose obtained in the lowly dynamic environment is relatively accurate, so most of the points are mapped to the correct position in the reference coordinate system of global point cloud map. That is, the quality of the global point cloud map obtained by the SLAM system in a lowly dynamic environment depends on whether the noise blocks are effectively removed.

In the sitting sequence, these two people are sitting on the chair all the time. From the left view of the global point cloud maps of Dyna-SLAM and DS-SLAM, we can clearly see the body contours of the two people. It can be seen from the fourth column of images that the noise blocks in the global point cloud map of our system are completely removed, that is, our system can effectively deal with the problem of map construction in a lowly dynamic environment.

When a robot uses the constructed map to navigate or interact with the environment, if there are more noise blocks in the map, it will inevitably have an adverse impact on the robot's decision. By comparing global point cloud maps constructed by the four SLAM systems in highly and lowly dynamic environments, we can see that the global point cloud maps of our SLAM system have advantages over the other three SLAM systems.

V. CONCLUSION

In this paper, we proposed a semantic SLAM system with more accurate point cloud map in dynamic environments. The bounding boxes and masks of the potential dynamic objects could be obtained with BlitzNet, and the image can be quickly divided into environment regions and dynamic regions by the bounding boxes. We introduce a novel statistical method of depth analysis to remove the noise blocks formed by the dynamic objects as well as the islands generated by geometric segmentation. We construct epipolar constraint by the depth-stable matching points in the environment regions, and the static matching points in the dynamic regions can be located by the constraint. The experimental results on five sequences of the TUM RGB-D dataset demonstrate that our method can eliminate the influence of the dynamic objects effectively. Comparisons with ORB-SLAM2, Dyna-SLAM and DS-SLAM show that our method has certain advantages in the accuracy of camera pose estimation and the integrity of the trajectory. To our knowledge, the global point cloud map constructed by our method looks the best among the maps built by the existing dynamic SLAM systems. Our system can effectively remove noise blocks from global point cloud maps in both highly and lowly dynamic environments, which is the main advantage of our system.

However, there are some shortcomings of the proposed method: Firstly, the potential dynamic objects are specified in advance based on life experience. If an unknown dynamic object occupies most of the camera's field of view, the system will regard the object as a part of the static environment regions, causing the camera's pose and trajectory estimation error. Secondly, the semantic information provided by BlitzNet is not fully utilized. Finally, we did not study the specific motion state of the dynamic object in the environment.

In view of the problems existing in the system, our future work includes: unknown dynamic object processing, construction of semantic map. At the same time, the robot is likely to collide with some dynamic objects when exploring the unknown environment. Therefore, we need to further study the motion of the dynamic objects in the environment to provide a safe navigation routes for the robot.

REFERENCES

[1] M. Adams, B.-N. Vo, R. Mahler, and J. Mullane, "SLAM gets a PHD: New concepts in map estimation," *IEEE Robot. Autom. Mag.*, vol. 21, no. 2, pp. 26–37, Jun. 2014.

[2] 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.

[3] 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. 1–10.

[4] R. Mur-Artal, J. M. 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, Oct. 2015.

[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, Oct. 2017.

[6] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis.*, Zurich, Switzerland, Sep. 2014, pp. 834–849.

[7] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.

[8] Y. Sun, M. Liu, and M. Q.-H. Meng, "Improving RGB-D SLAM in dynamic environments: A motion removal approach," *Robot. Auto. Syst.*, vol. 89, pp. 110–122, Mar. 2017.

[9] Y. Fan, H. Han, Y. Tang, and T. Zhi, "Dynamic objects elimination in SLAM based on image fusion," *Pattern Recognit. Lett.*, vol. 127, pp. 191–201, Nov. 2019.

[10] M. R. U. Saputra, A. Markham, and N. Trigoni, "Visual SLAM and structure from motion in dynamic environments: A survey," *ACM Comput. Surv.*, vol. 51, no. 2, pp. 37.1–37.36, 2018.

[11] 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)*, Honolulu, HI, USA, Jul. 2017, pp. 6243–6252.

[12] F. Zhong, S. Wang, Z. Zhang, C. Chen, and Y. Wang, "Detect-SLAM: Making object detection and SLAM mutually beneficial," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Lake Tahoe, NV, USA, Mar. 2018, pp. 1001–1010.

[13] N. Sunderhauf, T. T. Pham, Y. Latif, M. Milford, and I. Reid, "Meaningful maps with object-oriented semantic mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Vancouver, BC, Canada, Sep. 2017, pp. 5079–5085.

[14] X. Li and R. Belaroussi, "Semi-dense 3D semantic mapping from monocular SLAM," 2016, *arXiv:1611.04144*. [Online]. Available: http://arxiv.org/abs/1611.04144

[15] J. Civera, D. Galvez-Lopez, L. Riazuelo, J. D. Tardos, and J. M. M. Montiel, "Towards semantic SLAM using a monocular camera," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Francisco CA, USA, Sep. 2011, pp. 1277–1284.

[16] N. Dvornik, K. Shmelkov, J. Mairal, and C. Schmid, "BlitzNet: A real-time deep network for scene understanding," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Honolulu, HI, USA, Oct. 2017, pp. 4154–4162.

[17] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Vilamoura, Portugal, Oct. 2012, pp. 573–580.

[18] D. Zou and P. Tan, "CoSLAM: Collaborative visual SLAM in dynamic environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 354–366, Feb. 2013.

[19] C. Evers and P. A. Naylor, "Optimized self-localization for SLAM in dynamic scenes using probability hypothesis density filters," *IEEE Trans. Signal Process.*, vol. 66, no. 4, pp. 863–878, Feb. 2018.

[20] M. S. Bahraini, M. Bozorg, and A. B. Rad, "SLAM in dynamic environments via ML-RANSAC," *Mechatronics*, vol. 49, pp. 105–118, Feb. 2018.

[21] G. Liu, W. Zeng, B. Feng, and F. Xu, "DMS-SLAM: A general visual SLAM system for dynamic scenes with multiple sensors," *Sensors*, vol. 19, no. 17, p. 3714, Aug. 2019.

[22] J. Bian, W.-Y. Lin, Y. Matsushita, S.-K. Yeung, T.-D. Nguyen, and M.-M. Cheng, "GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 2828–2837.

[23] Y. Sun, M. Liu, and M. Q.-H. Meng, "Motion removal for reliable RGB-D SLAM in dynamic environments," *Robot. Auto. Syst.*, vol. 108, pp. 115–128, Oct. 2018.

[24] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Tokyo, Japan, Nov. 2013, pp. 2100–2106.

[25] R. Scona, M. Jaimez, Y. R. Petillot, M. Fallon, and D. Cremers, "StaticFusion: Background reconstruction for dense RGB-D SLAM in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Brisbane, QLD, Australia, May 2018, pp. 1–9.

[26] 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, Oct. 2017.

[27] 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.

[28] W. Dai, Y. Zhang, P. Li, and Z. Fang, ''RGB-D SLAM in dynamic environments using points correlations,'' 2018, *arXiv:1811.03217*. [Online]. Available: http://arxiv.org/abs/1811.03217

[29] H. Rebecq, G. Gallego, and D. Scaramuzza, ''EMVS: Event-based multi-view stereo,'' in *Proc. Brit. Mach. Vis. Conf.*, York, U.K., 2016, pp. 1–111.

[30] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, ''Events-to-video: Bringing modern computer vision to event cameras,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 3857–3866.

[31] H. Rebecq, G. Gallego, E. Mueggler, and D. Scaramuzza, ''EMVS: Event-based multi-view stereo—3D reconstruction with an event camera in real-time,'' *Int. J. Comput. Vis.*, vol. 126, no. 12, pp. 1394–1414, Dec. 2018.

[32] H. Rebecq, T. Horstschaefer, and D. Scaramuzza, ''Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization,'' in *Proc. Brit. Mach. Vis. Conf.*, London, U.K., 2017, pp. 1–8.

[33] H. Rebecq, T. Horstschaefer, G. Gallego, and D. Scaramuzza, ''EVO: A geometric approach to event-based 6-DOF parallel tracking and mapping in real time,'' *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 593–600, Apr. 2017.

[34] B. Bescos, J. M. Facil, J. Civera, and J. Neira, ''DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes,'' *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 4076–4083, Oct. 2018.

[35] M. Runz, M. Buffier, and L. Agapito, ''Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects,'' in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Munich, Germany, Oct. 2018, pp. 10–20.

[36] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger, ''MID-fusion: Octree-based object-level multi-instance dynamic SLAM,'' in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Montreal, QC, Canada, May 2019, pp. 5231–5237.

[37] 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.

[38] K. He, G. Gkioxari, P. Dollár, and R. Girshick, ''Mask R-CNN,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Venice, Italy, Oct. 2017, pp. 2961–2969.

[39] 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)*, Madrid, Spain, Oct. 2018, pp. 1168–1174.

[40] L. Cui and C. Ma, ''SOF-SLAM: A semantic visual SLAM for dynamic environments,'' *IEEE Access*, vol. 7, pp. 166528–166539, 2019.

[41] L. Cui and C. Ma, ''SDF-SLAM: Semantic depth filter SLAM for dynamic environments,'' *IEEE Access*, vol. 8, pp. 95301–95311, 2020.

[42] 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, Dec. 2017.

[43] N. Brasch, A. Bozic, J. Lallemand, and F. Tombari, ''Semantic monocular SLAM for highly dynamic environments,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Madrid, Spain, Oct. 2018, pp. 393–400.

[44] T. Sun, Y. Sun, M. Liu, and D.-Y. Yeung, ''Movable-Object-Aware visual SLAM via weakly supervised semantic segmentation,'' 2019, *arXiv:1906.03629*. [Online]. Available: http://arxiv.org/abs/1906.03629

[45] 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*, early access, Apr. 30, 2020, doi: 10.1109/ACCESS.2020.2991441.

[46] Q. Jin, Z. Meng, T. D. Pham, Q. Chen, L. Wei, and R. Su, ''DUNet: A deformable network for retinal vessel segmentation,'' *Knowl.-Based Syst.*, vol. 178, pp. 149–162, Aug. 2019.

[47] S. Yang, J. Wang, G. Wang, X. Hu, M. Zhou, and Q. Liao, ''Robust RGB-D SLAM in dynamic environment using faster R-CNN,'' in *Proc. 3rd IEEE Int. Conf. Comput. Commun. (ICCC)*, Qingdao, China, Dec. 2017, pp. 2398–2402.

[48] S. Ren, K. He, R. Girshick, and J. Sun, ''Faster R-CNN: Towards real-time object detection with region proposal networks,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[49] L. Zhang, L. Wei, P. Shen, W. Wei, G. Zhu, and J. Song, ''Semantic SLAM based on object detection and improved octomap,'' *IEEE Access*, vol. 6, pp. 75545–75559, 2018.

[50] J. Redmon and A. Farhadi, ''YOLO9000: Better, faster, stronger,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 7263–7271.

[51] L. Xiao, J. Wang, X. Qiu, Z. Rong, and X. Zou, ''Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment,'' *Robot. Auto. Syst.*, vol. 117, pp. 1–16, Jul. 2019.

[52] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, ''SSD: Single shot multibox detector,'' in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands, Oct. 2016, pp. 21–37.

[53] L. Sun, F. Kanehiro, I. Kumagai, and Y. Yoshiyasu, ''Multi-purpose SLAM framework for dynamic environment,'' in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Honolulu, HI, USA, Jan. 2020, pp. 519–524.

[54] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, ''OctoMap: An efficient probabilistic 3D mapping framework based on octrees,'' *Auto. Robots*, vol. 34, no. 3, pp. 189–206, Apr. 2013.

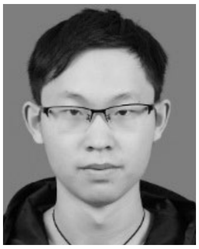
[55] M. A. Fischler and R. C. Bolles, ''Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,'' *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[56] X. Gao and T. Zhang, ''Robust RGB-D simultaneous localization and mapping using planar point features,'' *Robot. Auto. Syst.*, vol. 72, pp. 1–14, Oct. 2015.

YINGCHUN FAN received the B.S. degree in intelligence science and technology from the School of Electronic Engineering, Xidian University, Xi'an, China, in 2014, and the M.S. degree in optical engineering from the School of Physics and Information Technology, Shaanxi Normal University, Xi'an, in 2017. He is currently pursuing the Ph.D. degree in pattern recognition and intelligent system with the School of Artificial Intelligence, Xidian University.

QICHI ZHANG received the B.S. degree in electronics and information engineering from Shenzhen University, China, in 2017. He is currently pursuing the M.S. degree in electronics and communication engineering with the School of Artificial Intelligence, Xidian University, Xi'an, China. His research interests include computer vision, simultaneously localization and mapping, and machine learning, with a focus on dynamic slam.

SHAOFENG LIU received the bachelor's degree from Northwestern Polytechnical University, Xi'an, China, in 2017. He is currently pursuing the master' degree with the School of Artificial Intelligence, Xidian University. His research interests include computer vision, image processing, and object detection.



YULIANG TANG was born in 1994. He received the bachelor's degree from the Xi'an University of Science and Technology, Xi'an, China, in 2017. He is currently pursuing the master' degree with the School of Artificial Intelligence, Xidian University. His research interests include SLAM, computer vision, and semantic segmentation.

XIN JING was born in 1991. He received the master's degree in electronic engineering from Xidian University, Xi'an, China, in 2017. He works at the 20th Research Institute of China Electronica Technology Group Corporation, Xi'an. His research interests include intelligent navigation and computer vision.

JINTAO YAO received the master's degree from Xidian University, Xi'an, China, in 2018. He is currently an Engineer with the 20th Research Institute of China Electronica Technology Corporation, Xi'an. His research interests include integrated navigation and inertial navigation.

HONG HAN (Member, IEEE) was born in 1974. She received the Ph.D. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2003. She is currently a Senior Researcher with the School of Artificial Intelligence, Xidian University, Xi'an. Her research interests include computer vision, information fusion, and machine learning.