Visual SLAM Based on Semantic Segmentation and Geometric Constraints for Dynamic Indoor Environments

SHIQIANG YANG1, CHENG ZHAO1, ZHENGKUN WU2, YAN WANG1, GUODONG WANG1, AND DEXIN LI1

1School of Mechanical and Precision Instrument Engineering, Xi’an University of Technology, Xi’an 710048, China
2The Sixth Research Institute of China Aerospace Science and Industry Corporation 210, Xi’an 710048, China

Corresponding author: Shiqiang Yang (yangsq@126.com)

This work was supported by the National Natural Science Foundation of China under Grant 51475365.

ABSTRACT Simultaneous localization and mapping (SLAM), a core technology of mobile robots and autonomous driving, has received more and more attention in recent years. However, most of the existing visual SLAM algorithms do not consider the impact of dynamic objects on the visual SLAM system, resulting in significant system positioning errors and high map redundancy. Based on the ORB-SLAM2 algorithm, this paper combines semantic information and a geometric constraint algorithm based on feature point homogenization to improve the positioning accuracy of the SLAM system. Aiming at the problem that the feature points extracted by the ORB-SLAM2 algorithm are easily concentrated, and the extraction rate is low in the weak texture area, a feature point homogenization algorithm based on quadtree and adaptive threshold is proposed to improve the uniformity of feature point extraction. In addition, in view of the impact of dynamic targets on the SLAM system, the dynamic information in the scene is filtered out through the semantic segmentation network and the motion consistency detection algorithm, and the static feature points obtained after filtering are used to estimate the camera pose. Then, a semantic map is constructed after filtering out the dynamic point cloud according to the semantic information. Finally, the test results on Oxford and TUM datasets show that the uniformity of feature points extracted by the improved algorithm is increased by 56.3%. The positioning error of visual SLAM is reduced by 68.8% on average, and the constructed semantic map has rich semantic information and less redundancy.

INDEX TERMS Visual SLAM, ORB-SLAM2, dynamic feature filtering, semantic map.

I. INTRODUCTION

SLAM means that the mobile robot can complete its localization and environmental map construction without any prior knowledge of the environment [1]. VSLAM (Visual SLAM) perceives the environment through a camera, with low cost and rich image information, which has good development prospects.

According to different registration methods, VSLAM can be divided into direct method SLAM [2], [3] and feature point method SLAM [4]. The feature point method is less sensitive to illumination and dynamic objects than the direct method, so it is the current mainstream solution in VSLAM. Although the current VSLAM technology development has made remarkable achievements, there are still some problems to be solved. For example, ORB [5] feature points are based on FAST [6] key points, and feature point judgment is performed by comparing the brightness values of local regions. Therefore, the feature points extracted in regions with rich texture information are concentrated. Feature point extraction is the basis of VSLAM pose estimation, which affects the positioning accuracy of the VSLAM system. Therefore, feature point extraction should be evenly distributed. Otherwise, the extracted feature points cannot reflect global information. In the weak texture area, the image information is close. How to effectively increase the number of feature points in this area is a significant problem in VSLAM research, and the traditional extraction method using empirical threshold is less robust. In addition, most VSLAM system based on an assumption of a static environment does not consider the...
influence of dynamic objects. Introducing these dynamic feature points reduces the system’s positioning accuracy when performing pose estimation. Map construction is the core problem of VSLAM system. However, the constructed map has much redundancy in a dynamic environment, and readability is reduced due to dynamic objects. How to effectively improve the uniformity of feature point distribution and reduce the impact of dynamic targets on mobile robots is a significant challenge facing the current VSLAM system.

In order to solve the above problems, this paper proposes several solutions. The main contributions are summarized as follows:

1) An algorithm based on quadtree homogenization and an adaptive threshold is proposed to improve the uniformity of feature point extraction and the number of feature points extracted in weak texture areas.
2) A dynamic feature point removal algorithm is proposed. The algorithm mainly uses the semantic segmentation network and a geometric constraint method to filter out the dynamic features in the environment and improve the positioning accuracy of VSLAM algorithm in a dynamic environment.
3) A construction method of point cloud and a semantic map is proposed, which can effectively remove the dynamic point cloud information in the scene and build a semantic map.

The rest of this paper is structured as follows. Section II discusses related work. Section III describes a system overview and details the implementation of the proposed methods. Section IV shows experimental results, and Section V presents the conclusions and discusses future work.

II. RELATED WORK

Despite facing many obstacles, there is still motivation to find a novel VSLAM algorithm for indoor dynamic scenes to improve the performance of VSLAM. Qtree_ORB algorithm proposed by Mur-Artal R [7], [8] et al., improves the uniformity of ORB feature points to a certain extent by introducing the method of quad-tree homogenization. However, the extraction rate of feature points in weak texture areas is low. Geng C [9] et al., adopted an adaptive threshold method to improve the feature point extraction rate in weak texture areas, but the problem of low feature point uniformity did not change. UR-SIFT [10] algorithm extracted by Paul S et al., significantly improves the uniformity of feature point distribution to obtain better matching performance.

To reduce the impact of dynamic objects on VSLAM system, more and more scholars have carried out research on this. Sun [11] used the difference between adjacent frames to detect dynamic points, but this method will cause motion blur when the camera is moving, so the detection effect is poor. Azartash [12] distinguished moving regions from static backgrounds by segmenting images from the RGB-D sensor. Kim [13] obtained static regions of an image by computing depth difference between consecutive frames. Wang [14] et al., identified dynamic points in a scene by combining epipolar geometric constraints and depth map clustering information. Li and Lee [15] proposed a static weighting method for keyframe edge points based on the RGB-D SLAM system, which distinguishes static and dynamic points according to the static weight judgment. Dyna-SLAM [16] proposed by Bescós et al., this algorithm obtains potential dynamic prior information in the environment by introducing Mask-RCNN [17] network and then filters out dynamic feature points with geometric constraints, it can effectively improve the accuracy of pose estimation. Dynamic-SLAM [18] and Detect-SLAM [19], detected dynamic objects and filtered them out through SSD [20] network. DDL-SLAM [21] used DU-Net [22] to detect prior dynamic information and combined depth and geometric information to filter out dynamic feature points. OD-SLAM [23] proposed by Xu H et al., this algorithm uses Seg-Net [24] network to obtain dynamic target information in the scene, and filters out the dynamic features by combining the reprojection error. However, the accuracy of the semantic segmentation is low so recognition may fail.

As the SLAM system’s core problem, building a clear and reusable map is a significant challenge. Lai L [25] et al., combined PSP-Net [26] network to predict the semantic category of each pixel and finally obtained an octree map with semantic information. However, it does not filter out dynamic objects in the scene, so the obtained map has a high overlap rate. Q. Ju [27] et al., established a point cloud map with semantic information based on ORB-SLAM2 combined with the YOLOv5 target detection algorithm and used the VCCS (Voxel Cloud Connectivity Segmentation) algorithm for super-voxel clustering, but the algorithm has low accuracy. Menini [28] proposed an online method to jointly infer the 3D structure and 3D semantic labels of indoor scenes in real-time, but the accuracy of this algorithm is related to the resolution of the depth map, and the accuracy of the constructed map is reduced when resolution is lower.

III. SYSTEM DESCRIPTION

A. SYSTEM FRAMEWORK

ORB-SLAM2 system consists of a tracking thread, a local map construction thread, and a loop closure detection thread. The tracking thread calculates the camera’s motion pose through extraction and matching feature points, the local map thread constructs a sparse point cloud map, and the loop detection thread performs loop detection through the bag of words model [29] to reduce the cumulative error of the system. This paper adds a new semantic segmentation thread and a static semantic map construction thread to obtain semantic information and build semantic maps. The overall architecture of the improved system is shown in Fig. 1, where the blue area represents the improved or increased modules. The main improvements are as follows.
1) TRACKING THREAD
In this thread, the feature point extraction module of the ORB-SLAM2 algorithm has been improved to increase the uniformity of feature point extraction and feature point extraction rate in weak texture areas.

2) SEMANTIC SEGMENTATION THREAD
As a new module, this thread mainly obtains semantic information in the scene through the semantic segmentation network DPT [30].

3) SEMANTIC MAP CONSTRUCTION THREAD
This module obtains semantic information by combining semantic segmentation threads and drawing keyframes through co-view relationship and similarity detection algorithm. Then, a static 3D point cloud map and a semantic map will be built according to the motion pose of the camera.

B. IMPROVED ORB FEATURE EXTRACTION
ORB feature draws the advantages of FAST detection and Brief [31] descriptor. It builds a multi-layer feature point extraction model with high real-time performance and good rotation and size invariance.

The extraction principle of FAST key points is shown in Fig. 2. FAST is a kind of corner that determines whether a point is a key point by detecting the change in the gray value of the local pixel. If a pixel is significantly different from the pixels in its neighborhood, this point may be a key point. Therefore, ORB feature extraction relies on the bright information of local pixels and threshold size, making ORB feature points easy to gather in local areas with strong texture information.

The improved feature point extraction algorithm is shown in Table 1. First, an image pyramid based on the input RGB image is built to obtain multi-resolution image scale information. Then feature points are extracted for each layer of pyramid image, where the number of the layers is set to
an empirical value of 8, and the number of feature points extracted from each layer is:

$$\text{DesF}_i = N \times \frac{1 - \text{InvSF}_i}{1 - \text{InvSF}_j} \times \text{InvSF}_i$$

In the formula, $N$ is the total number of expected feature points, $N_i$ represents the total number of layers in the pyramid, $\text{DesF}_i$ represents the number of feature points that need to be extracted for layer $i$ of the pyramid, and $\text{InvSF}$ is the reciprocal of the scaling factor of each pyramid image. The gray value of the image in the weak texture area is relatively close, and the effect of using a fixed threshold for feature point extraction is poor. Therefore, according to the distribution of current grid gray value to calculate feature point extraction threshold value of this area. The threshold calculation method is as follows.

$$T = \sum_{u,v \in (0, M)} I(u, v)$$

$$\min T_h = \frac{1}{k} \sum_{u,v \in (0, M)} (T - I(u, v))^2$$

In the formula, $M$ represents the size of current grid image, $k$ is the scale factor and $I(u, v)$ represents gray value at $u,v$ in the pixel coordinate system.

**TABLE 1. Feature point homogenization algorithm.**

| Algorithm 1 Feature Extraction Algorithm Based on Quadtree Homogenization |
|--------------------------------------------------|
| **Input:** Image frame | **Output:** ORB feature points |
| 1. build image pyramid | 2. for every pyramid layer $l$ do |
| 3. ORB feature point extraction by adaptive threshold | 4. divide quadtree nodes on level $l$ pyramid |
| 5. for every child node $j$ do | 6. if $\text{Num}(l) > 1 \& \& \& \text{noFlag}(l)$ do |
| 7. divide child quadtree nodes in parent $j$ node | 8. else |
| 9. $\text{noFlag}(l)$ = true | 10. end if |
| 11. retain features with large Harris response values | 12. filter out feature points whose Euclidean distance is less than a threshold $\lambda$ |
| 13. end for | 14. end for |

Then, a quad-tree is constructed for feature point homogenization. When the number of nodes is greater than the target number or the feature number of the current node is equal to 1, stop the quad-tree division. If the termination condition is reached, the feature point with the largest Harris response value is retained for nodes with more than two feature points. Finally, the feature points whose distance is less than the threshold are filtered out according to the Euclidean distance between the extracted feature points.

**C. DYNAMIC FEATURE POINT FILTERING**

Dynamic feature points introduce significant errors to the precision estimation of VSLAM. How to effectively filter out these dynamic points has become a hot topic in VSLAM research. Based on the semantic segmentation network, this paper realizes the detection and filtering of dynamic feature points in the scene by combining it with a dynamic consistency detection algorithm.

1) **SEMANTIC SEGMENTATION**

In this section, DPT semantic segmentation network is used to detect the prior dynamic information in the scene and provide semantic constraint information for further algorithms to filter out dynamic feature points. Unlike traditional neural networks, the backbone of DPT is a visual transformer rather than a convolutional network, which can process higher-resolution images at a relatively constant speed. This dense visual transformer enables finer and more globally coherent predictions than fully convolutional neural networks.

The prior dynamic target information in the scene can be obtained through the semantic segmentation network, which simplifies the identification and removal of dynamic feature points. However, most semantic segmentation networks have low semantic segmentation accuracy. In complex dynamic environments, the collected images have motion blur and cannot obtain prior dynamic information. Combined with geometric information, the detection of dynamic feature points is realized to make up for the lack of semantic information. Traditional methods use optical flow to track dynamic targets, which are easily affected by illumination changes and have poor noise immunity. The dynamic point detection by projection error depends on the accurate fundamental matrix. The existence of dynamic objects usually does not destroy the relative positional relationship between static objects in space, and the feature points are less sensitive to illumination changes. In view of this, a dynamic point detection algorithm based on spatial structure relationship is extracted, and the specific algorithm is as follows.

2) **MOTION CONSISTENCY DETECTION**

As shown in Fig. 3, point $O_1$ is the location of camera, $A$, $B$ and $C$ are three points in space, after $\Delta t$ time, the camera moves from $O_1$ to $O_2$, and the corresponding points are $A'$, $B'$, $C'$, setting the corresponding edge length of $\Delta ABC$ to $d_{AB} = d_{B1}$, $d_{AC} = d_{B2}$, $d_{BC} = d_{B3}$, the corresponding lengths of the three sides of $\Delta A'B'C'$ are $d_{A'B'} = d_{q_1}$, $d_{A'C'} = d_{q_2}$, $d_{B'C'} = d_{q_3}$. When there are no dynamic objects in the scene, the extracted feature points are all static feature points, and the corresponding margin values of space $\Delta ABC$ and $\Delta A'B'C'$ should be within a small range.

$$|d(p_1, p_j, p_k) - d(q_1, q_j, q_k)| \in (d_{\text{min}}, d_{\text{max}})$$
In order to better describe this constraint relationship, define the following constraint equation:

$$q(i, j, k) = \exp\left(\frac{|d(p_i, p_j, p_k) - d(q_i, q_j, q_k)|}{A(i, j, k)}\right)$$

In the formula, $A(i, j, k)$ represents the Euclidean distance between two points in space, which is defined as follows:

$$A(i, j, k) = \frac{|d(p_i, p_j, p_k) + d(q_i, q_j, q_k)|}{2}$$

Secondly, since the RGB-D camera is used, depth information can be obtained. If there is a dynamic object in the space, the depth value of this corresponding feature point will change abruptly. The depth change equation is defined as follows:

$$D(i, j, k) = |D_p(i, j, k) - D_q(i, j, k)|$$

### D. SEMANTIC MAP CONSTRUCTION

The perception of the environment depends on the geometric, three-dimensional structure and semantic understanding of the environment. As the perception carrier of geometry and semantics, the 3D semantic map has gradually received attention. The main process of building a semantic map is shown in Fig. 4.

The main work of constructing the 3D semantic map algorithm is as follows.

1) Deep information transformation. The depth information of each pixel is obtained according to the depth map. Due to the existence of measurement error, it is necessary to filter the depth image and then map the depth value to the three-dimensional space according to the internal parameter matrix of the depth camera.

2) Color information acquisition. The point cloud color information is obtained from the RGB image, and the keyframe point cloud map is constructed according to the key selection strategy.

3) Dynamic target filtering. Due to the existence of dynamic targets in dynamic scenes, there will be a lot of redundant information in the constructed static map, which reduces the readability of the map. This paper filters out the dynamic objects in the scene through semantic information.

4) Loop-back judgment. Loop-back detection is performed on the input mapping keyframe. If it is a loop-back, the global map is updated. Otherwise, insert this keyframe.

5) Semantic map construction. Combine semantic information to build a semantic map.

Using keyframes have a good effect in constructing a sparse point cloud map, but it will lead to redundant map information when building a dense point cloud map. Therefore, it is necessary to filter the newly inserted keyframes and use drawing keyframes instead of keyframes for point cloud map construction. The keyframe selection strategy is as follows:

1) Common vision relationship detection: First, all keyframe map points are saved. When a new keyframe is inserted, compare whether the map point in the newly inserted keyframe is included in the saved map points. If only a part of the key points is included, it means that there is a less common observation relationship between the keyframe and keyframes in the current map, then insert this keyframe.

2) A keyframe information base is established, and when a new keyframe is inserted, the similarity between the keyframe and the information base is calculated, and a high similarity indicates that the current frame has more redundant information. The minimum similarity threshold is set to $\lambda_{\text{min}}$ and keep the keyframe when the similarity is less than $\lambda_{\text{min}}$.

The similarity detection is based on the bag-of-words model. When there is a newly inserted keyframe, the feature calculation of the newly inserted keyframe is performed to obtain the feature vector of the current frame, and the similarity is calculated by comparing it with the feature vector existing in the dictionary. The similarity calculation adopts
S. Yang et al.: Visual SLAM Based on Semantic Segmentation and Geometric Constraints for Dynamic Indoor Environments

FIGURE 5. Uniformity test results under the Oxford dataset. Figure (a) is the uniform distribution curve, Figure (b) is the correct matching point curve, Figure (c) is the matching correct rate curve.

TABLE 2. Distribution uniformity of feature points in the dataset.

| Algorithm      | bark  | brick | graf  | bike  | car   | boat  | ubc   | trees |
|----------------|-------|-------|-------|-------|-------|-------|-------|-------|
| Std_ORB        | 139.6 | 64.52 | 111.37| 117.69| 104.42| 92.39 | 85.59 | 94.84 |
| Qtree_ORB      | 72.47 | 99.06 | 47.8  | 40.63 | 37.16 | 92.91 | 82.69 | 91.34 |
| Improved_ORB   | 26.65 | 27.55 | 38.6  | 54.42 | 42.37 | 47.84 | 67.24 | 34.17 |

the TF-IDF [32] model:

\[
IDF_i = \log \frac{n}{n_i}
\]

In the formula, \(n\) represents the number of all features in the dictionary, and \(n_i\) represents the number of a feature in the dictionary.

Assuming that a feature \(o_i\) in an image appears \(n_i\) times, and the number of features that appear in the image is \(n\), then:

\[
TF_i = \frac{n_i}{n}
\]

The VSLAM system constructs a map by perceiving the external environment through a camera, and drawing keyframes are used as the information input for map construction, which plays a crucial role in map construction. However, since the drawing keyframes contain dynamic moving objects, if the dynamic information is not processed, the constructed environment map will have a lot of overlapping redundant data, which is not conducive to reuse of the map. This paper detects dynamic map points in the scene based on semantic information and filters out the map points marked as dynamic information.

IV. EXPERIMENTS AND RESULTS

A. UNIFORMITY EXPERIMENTAL TEST

Verify the uniformity of feature point distribution by calculating the standard deviation of image feature point distribution [33]. The smaller the standard deviation, the more uniform the distribution of feature points. All experiments are based on a computer with a processor model of i5-9400, a memory size of 8G, and an operating system of Ubuntu16.04.

Tested the extraction and matching performance of the improved algorithm feature points on the Oxford dataset, which includes scale change (bark/boat), blur change (bicycle/tree), rotation change (bark/boat), illumination change (car), and perspective changes (brick/graf), and compression changes (ubc). In order to verify the performance of the algorithm, the improved ORB algorithm is compared with the Qtree_ORB algorithm in ORB-SLAM2 and the ORB algorithm in the open-source image processing library OpenCV (3.3.1). The algorithm marked OpenCV is Std_ORB, and the algorithm in this paper is Improved_ORB.

The uniform degree of the three algorithms is calculated in Table 2 and Fig. 5(a). The Improved_ORB algorithm has the lowest standard deviation of uniformity compared with the ORB algorithm and Qtree_ORB, it average uniformity is reduced by 56.3% and 23.5%, respectively. It can be seen from Fig. 5(b) that the ORB algorithm has the lowest number of correct matching points, Qtree_ORB and Improved_ORB algorithms have higher matching points. Under the changes of blur, illumination and zoom, the number of correct matching points of the improved algorithm is significantly increased, and the increase in the number of matching point pairs can reduce the tracking loss and improve the system’s robustness. It can be seen from Fig. 5(c) that all three algorithms have good matching accuracy.

B. DYNAMIC FEATURE POINT FILTERING TEST

The positioning accuracy of the improved algorithm was tested on the public dataset TUM. TUM dataset includes several typical SLAM schemes such as Handheld SLAM and Robot SLAM. This paper is aimed at indoor dynamic motion scenes, so the dynamic objects dataset in the TUM dataset
S. Yang et al.: Visual SLAM Based on Semantic Segmentation and Geometric Constraints for Dynamic Indoor Environments

FIGURE 6. High dynamic scene feature points filter out test results.

TABLE 3. Absolute estimation error test results for highly dynamic scenes (ATE[m]).

| Sequence   | ORB_SLAM2 | Improved | Error reduction rate |
|------------|-----------|----------|---------------------|
|            | RMSE      | STD      | RMSE                | STD      | RMSE    | STD      |
| w_half     | 0.6067    | 0.2719   | 0.0285              | 0.0150   | 95.30%  | 94.50%   |
| w_rpy      | 0.7586    | 0.3901   | 0.0289              | 0.0158   | 96.19%  | 95.96%   |
| w_static   | 0.3930    | 0.1529   | 0.0169              | 0.0086   | 95.69%  | 94.39%   |
| w_xyz      | 0.6594    | 0.3300   | 0.0137              | 0.0067   | 97.93%  | 97.98%   |

TABLE 4. Relative estimation error test results for high dynamic scenes (RPE[m]).

| Sequence   | ORB_SLAM2 | Improved | Error reduction rate |
|------------|-----------|----------|---------------------|
|            | RMSE      | STD      | RMSE                | STD      | RMSE    | STD      |
| w_half     | 0.3903    | 0.3010   | 0.0265              | 0.0127   | 93.21%  | 95.80%   |
| w_rpy      | 0.4455    | 0.3343   | 0.0370              | 0.0208   | 91.70%  | 93.79%   |
| w_static   | 0.2211    | 0.1987   | 0.0141              | 0.0080   | 93.61%  | 95.97%   |
| w_xyz      | 0.3830    | 0.2288   | 0.0185              | 0.0091   | 95.17%  | 96.04%   |

was used for the test. And according to the movement speed of the characters in the dataset scene, the dataset is divided into two categories: low dynamic motion scene (sitting) and high dynamic motion scene (walking). Each scene contains several typical camera motion modes, halfsphere, r-p-y, static, x-y-z. Among them, halfsphere means that the camera moves along the hemisphere trajectory, r-p-y means that the camera can rotate along the x-y-z axis, and static means that the camera is basically fixed. The name in the figure adopts the format of “dataset name+ algorithm type”, and simplifies the high dynamic scene walking to w, low dynamic scene sitting to s.

Using the absolute trajectory error (ATE) and relative pose error (RPE) to measure the positioning accuracy of the algorithm. Since the robustness and stability of the system are closely related to root mean square error (RMSE) and standard deviation (STD), this paper uses RMSE and STD as the error metrics.

1) HIGH DYNAMIC SCENE SLAM POSITIONING ACCURACY TEST AND RESULT ANALYSIS

The filtering results of dynamic feature points in high dynamic scenes are shown in Fig. 6. It can be seen from Fig. 6(a)(c)(e)(g) that a large part of the extracted feature points come from the characters in the scene. These are potential dynamic feature points and have an important impact on the pose estimation of the system. Fig. 6(b)(d)(f)(h) are the feature points extracted by the algorithm in this paper. It can be seen from the figure that the extracted feature points are evenly distributed, and most of the dynamic feature points have been filtered out. Even if the camera is rotating or tilting, it can filter out dynamic feature points well, such as Fig. 6(b) and Fig. 6(d).

The test results of absolute and relative trajectory errors in high dynamic scenarios are shown in Tables 3 and 4. It can be seen that the positioning error of the system can...
be significantly reduced by filtering out the dynamic feature points. This is because the PnP [34] pose estimation relies on the feature points extracted from the scene. Due to dynamic feature points, the matching relationship between frames is destroyed, resulting in a significant estimated pose error. The relative trajectory error calculates the pose transformation relationship between adjacent frames. The relationship between the two frames is close, and the presence of dynamic features has a significant impact on pose estimation between the two frames. By filtering out the dynamic feature points in the scene, the relative trajectory error of the system can be effectively reduced, and the degree of drift of the system can be reduced.

The comparison test results of the camera’s actual trajectory and the algorithm estimated trajectory error in the high dynamic scene as shown in Fig. 7. Among them, the black line represents the actual trajectory of the camera, the blue line represents the estimated trajectory, and the red line represents the error value between the actual trajectory and the estimated trajectory. The longer the red line, the more significant error between the actual value and estimated value. It can be seen from the figure that the error value between the actual trajectory and the estimated trajectory of the ORB-SLAM2 algorithm is significant. After filtering out the dynamic feature points, the error value between the two is significantly reduced. Especially on the w_xyz and w_half datasets, the actual trajectory and the algorithm estimated trajectory almost coincide, indicating that the improved algorithm is effective.

2) LOW DYNAMIC SCENE SLAM POSITIONING ACCURACY TEST AND RESULT ANALYSIS

The filtering results of dynamic feature points in low dynamic scenes are shown in Fig. 8. From Fig. 8(a)(c)(e)(g), it can
TABLE 5. Absolute trajectory error test results for low dynamic scenes (ATE [m]).

| Sequence | ORB_SLAM2 | Improved | Error reduction rate |
|----------|-----------|----------|----------------------|
|          | RMSE      | STD      | RMSE              | STD      | RMSE   | STD     |
| s_half   | 0.0395    | 0.0161   | 0.0182            | 0.0088   | 53.93% | 44.92%  |
| s static | 0.0092    | 0.0043   | 0.0065            | 0.0036   | 28.88% | 16.45%  |

TABLE 6. Relative trajectory error test results for low dynamic scenes (RPE [m]).

| Sequence | ORB_SLAM2 | Improved | Error reduction rate |
|----------|-----------|----------|----------------------|
|          | RMSE      | STD      | RMSE              | STD      | RMSE   | STD     |
| s_half   | 0.0297    | 0.0219   | 0.0211            | 0.0105   | 29.08% | 51.82%  |
| s static | 0.0091    | 0.0042   | 0.0083            | 0.0043   | 7.84%  | -2.38%  |

FIGURE 9. Comparison test results of the camera’s actual trajectory and the algorithm’s estimated trajectory error in low dynamic scenes.

be seen that the distribution of the extracted feature points is relatively uniform, and the number of extracted feature points is relatively abundant. However, many feature points are gathered on potential dynamic targets. The improved algorithm can effectively remove potential and dynamic feature points, but there are certain errors in the algorithm under individual datasets. For example, in Fig. 8(b), (c), and (f), there are a small number of dynamic points that exist near the right foot of the character in the scene. On the one hand, it is due to the insensitivity to low-moving objects based on the motion consistency constraint algorithm. On the other hand, the edges of the segmentation are not precise due to the influence of the accuracy of the semantic segmentation algorithm. In Fig. 8(d), due to the complex motion of the camera, the extracted RGB image is blurred, so there is an error in the filtering of dynamic feature points.

The test results of absolute and relative trajectory errors in low dynamic scenes are shown in Tables 5 and 6. In the low dynamic scene, most of the feature points tend to be static, and only a few are dynamic. Since the number of dynamic feature points is small in low dynamic scenes, the pose matrix is less affected during motion estimation, resulting in a smaller absolute trajectory error value. Similarly, the relative error values in Table 6 are also small, and the standard deviation in the static dataset has increased. This is due to the relative trajectory error being evaluated by evaluating the pose transformation between adjacent frames, which is less sensitive to low dynamic motion during the evaluation process. As a result, some dynamic feature points are not filtered out in time, impacting the correct information association of consecutive frames and increasing the error. Fig. 9 shows the comparison test results of the camera’s actual trajectory and the algorithm’s estimated trajectory error. It can be seen from the figure that the small number of dynamic points in low dynamic scenes has little effect on the positioning accuracy of the ORB-SLAM2 algorithm. After filtering out the dynamic points, the improved algorithm can improve the positioning accuracy of VSLAM and reduce the impact of dynamic feature points.

C. SEMANTIC MAP CONSTRUCTION TEST

Tested the effectiveness of semantic map construction on the TUM dataset. In order to facilitate the description of this paper, the algorithm for map construction through keyframes is marked as KM, the improved algorithm of filtering through drawing keyframes and dynamic map points is marked as IKM, and the semantic map is marked as SM. Similarly, the dynamic map points are filtered out when the semantic map is constructed. The naming format in the figure is consistent with the previous section.

The test results of the point cloud map and semantic map construction under the TUM dataset are shown in Fig. 10. It can be seen that the point cloud map constructed by
keyframes has a lot of redundant information, the point cloud overlap is high, and the map readability is poor. However, after the improved algorithm adopts the strategy of drawing key frames and filtering dynamic map points, the constructed map does not contain dynamic point clouds, with a clear spatial structure and less information redundancy. The constructed semantic map can effectively distinguish different objects in the scene with rich semantic information and high precision. Due to the accumulated error in the system, the constructed semantic map is partially overlapped during point cloud splicing. In the sitting_static dataset, the area where the dynamic target is located is partially missing after the dynamic map points are removed. This is because, in low dynamic scenes, the moving target and the camera do not move in a large range. Therefore, the point cloud of the following static background areas cannot be obtained so that the area cannot be inpainted.

D. ACTUAL ENVIRONMENT TEST

This paper tested the performance of the improved algorithm in a natural environment. In the experiment, the turtlebot mobile robot platform was used for data acquisition, the visual sensor was Kinect v1, and the operating system was Ubuntu 16.04. During the experiment, the camera kept moving in a straight line or a curve, the moving speed ranged from 0-0.5m/s, and the rotation speed ranged from 1-10rad/s. Each scene contains 1-3 dynamic targets, and the moving speed range is 0-1m/s.

The experimental test platform is shown in Fig. 11(a). This paper selects the office scene Fig.11 (b) and the corridor scene Fig.11 (c) to test the improved algorithm. The office scene has rich texture information and high complexity, so it has a strong representation. The corridor scene usually has weak texture information and a single scene, which is easily lost during feature extraction and can be used to test the robustness of system tracking. During the experiment, two frames were randomly selected for comparison:

1) FEATURE POINT HOMOGENIZATION EXPERIMENTAL TEST

The uniformity test results of the improved algorithm in the office scene and the corridor scene are shown in Fig. 12. From Fig. 12 (a) (b) (c) (d), it can be seen that the distribution of feature points extracted by quad-tree homogenization is relatively uniform. However, the ORB-SLAM2 algorithm is easy to gather in areas with rich texture information, such as in Fig. 12(a)(c) red wireframe area. By adopting an adaptive threshold, the improved algorithm can extract more feature points in areas with weak texture information, such as the blue wireframe area. Fig. 12 (e) (f) (g) (h) is a weak
texture scene. The wall and the ground texture are single, and when the number of extracted feature points is small, the system tracking is easy to lose. In the figure, the ORB-SLAM2 algorithm extracts fewer feature points in the area close to the ground. In comparison, the improved algorithm has high flexibility through the adaptive threshold algorithm and can obtain more feature points on the ground. In general, the feature points extracted by the improved algorithm are evenly distributed, improving the feature point extraction rate in the texture area.

2) EXPERIMENTAL TEST OF DYNAMIC FEATURE POINT FILTERING
The test results of dynamic feature point filtering in the office and corridor scene are shown in Fig. 13. From the comparison of Fig. 13(a) (b) and Fig. 13(c) (d), it can be seen that the extracted feature points are evenly distributed but also contain a large number of dynamic feature points. The improved algorithm can eliminate potential dynamic feature points in the scene through semantic information and motion consistency detection. However, there are a small number of
potential feature points that have not been filtered out. For example, in the feature points of the human head area in the scene, due to the far viewing angle, there is an error in the semantic segmentation, and the motion change is small when the motion consistency detection is passed. The sensitivity is low, so there is a certain error. Similarly, by comparing Fig. 13(e) (f) and (g) (h), it can be seen that even in scenes with weak texture information, the extracted feature points are evenly distributed, and dynamic feature points can be effectively filtered out.

3) MAP BUILD TEST
Fig. 14 shows the test results constructed for the point cloud map and semantic map of the office and the corridor scene. As seen in Fig. 14, the point cloud map built from keyframes contains many dynamic targets, resulting in the redundancy of map information. The improved algorithm can filter out the redundant point cloud information well, and the constructed map has a clear structure. The constructed semantic map has rich semantic information and high precision and can effectively distinguish objects such as the ground, walls, computer monitors, tables, and swivel chairs in the scene. In the corridor scene, the constructed point cloud map and semantic map do not contain dynamic objects, and the semantic map can be effectively constructed.

V. CONCLUSION
In this paper, a VSLAM algorithm based on semantic segmentation and motion consistency detection of feature point homogenization is proposed to reduce the influence of dynamic feature points on the positioning accuracy of the VSLAM system in the scene. Based on the ORB-SLAM2 algorithm framework, this paper improves the ORB feature extraction and map construction modules. The quadtree homogenization algorithm and geometric constraint algorithm are used to obtain feature points that do not contain dynamic information and are evenly distributed. Combined with semantic information, dynamic point cloud information is filtered out, and a semantic map is constructed. Experiments are performed on the Oxford dataset and TUM dataset, and the following conclusions are obtained:

1) The test results on the Oxford dataset show that the improved algorithm can effectively improve the feature point extraction rate in weak texture areas, and the uniformity of the extracted feature points is increased by 56.3%.

2) The proposed dynamic feature point filtering algorithm can effectively filter out the dynamic feature points in the scene and improve the positioning accuracy of the VSLAM system. The root mean square error of absolute trajectory error is reduced by 96.28% and 41.41% on average in the high-dynamic and low-dynamic scenes of the TUM dataset.

3) The constructed point cloud and semantic map do not contain dynamic point cloud information, the information redundancy is small, and the semantic information is rich.

Although the improved algorithm can effectively improve the positioning accuracy of the VSLAM system in dynamic environments, it has the following shortcomings:

1) The accuracy of the SLAM system is related to the extraction and matching of ORB feature points.
The system tracking is easily lost when the camera produces motion blur or the illumination changes. In the future, camera pose estimation can be combined with deep learning models.

2) Semantic information detection relies on the semantic segmentation network, which has high model complexity and a large amount of computation. In the future, the network model structure can be optimized to reduce the amount of computation.

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] B. Simsek, S. Satir, and H. S. Bilge, “Performance comparison of direct and feature based vSLAM algorithms,” in Proc. 29th Signal Process. Commun. Appl. Conf. (SIU), Jun. 2021, pp. 1–4.

[3] W. Zhang, Z. Yan, G. Xiao, A. Feng, and W. Zuo, “Real-time direct monocular SLAM with learning-based confidence estimation,” IEEE Access, vol. 7, pp. 103365–103376, 2019.

[4] B. Bescos, C. Campos, J. D. Tardos, and J. Neira, “DynaSLAM II: Tightly-coupled multi-object tracking and SLAM,” IEEE Robot. Autom. Lett., vol. 6, no. 3, pp. 5191–5198, Jul. 2021.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] C. Geng, J. Yang, J. Lin, T. Yu, and K. Shi, “An improved ORB feature extraction algorithm,” J. Phys., Conf., vol. 1616, no. 1, Aug. 2020, Art. no. 012026.

[10] S. Paul and U. C. Pati, “Remote sensing optical image registration using modified uniform robust SIFT,” IEEE Geosci. Remote Sens. Lett., vol. 13, no. 9, pp. 1300–1304, Sep. 2016.

[11] Y. Sun, M. Liu, and M. Q.-H. Meng, “Improving RGB-D SLAM in dynamic environments: A motion removal approach,” Robot. Autom. Syst., vol. 89, pp. 110–122, Mar. 2016.

[12] H. Azarhchi, K.-R. Lee, and T. Q. Nguyen, “Visual odometry for RGB-D cameras for dynamic scenes,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2014, pp. 1280–1284.

[13] 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. 1563–1573, Dec. 2016.

[14] 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.

[15] 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.

[16] B. Bescos, J. M. Fácil, J. Civera, and J. L. Neira, “DynaSLAM: Tracking, mapping, and inpainting in dynamic environments,” IEEE Robot. Autom. Lett., vol. 3, no. 4, pp. 4076–4083, Oct. 2018.

[17] Y. Song, G. Chen, and J. Liu, “An improved mask R-CNN with global context modeling for instance segmentation,” in Proc. Chin. Autom. Congr. (CAC), Nov. 2019, pp. 5644–5648.

[18] 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. Auton. Syst., vol. 117, pp. 1–16, Jul. 2019.

[19] 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. (WCAC), Mar. 2018, pp. 1001–1010.
CHENG ZHAO was born in Xi’an, Shaanxi, China, in 1995. He received the B.S. degree in mechanical design, manufacturing and automation from the Xi’an University of Technology, China, where he is currently pursuing the M.S. degree. His research interests include visual SLAM and image processing.

ZHENGKUN WU was born in Lintong, Shaanxi, China, in 1981. He is currently an Assistant Engineer at the 210 Institute of the Sixth Research Institute of China Aerospace Science and Industry Corporation. His main research interests include mechatronics and industrial control.

YAN WANG was born in Xianyang, Shaanxi, China, in 1998. She received the B.S. degree in mechanical design, manufacturing and automation from the Xi’an University of Technology, China, where she is currently pursuing the M.S. degree in mechanical and electronic engineering. Her research interests include visual SLAM and image processing.

GUODONG WANG was born in Xi’an, Shaanxi, China, in 1997. He received the B.S. degree in mechanical design, manufacturing and automation from the Xi’an University of Technology, in 2020, where he is currently pursuing the M.S. degree in mechanical engineering. His current research interests include laser and vision fusion SLAM.

DEXIN LI was born in Yantai, Shandong, China, in 1965. He received the Ph.D. degree in mechanics from Xi’an Jiaotong University, Xi’an, China, in 2003. From 1992 to 2019, he was with the Xi’an University of Technology, Xi’an. Since 2002, he has been an Associate Professor with the School of Mechanical and Precision Instrument Engineering, Xi’an University of Technology. His current research interests include mechanical computer-aided design and manufacturing.