MGC-VSLAM: A Meshing-Based and Geometric Constraint VSLAM for Dynamic Indoor Environments

SHIQIANG YANG, GUOHAO FAN, LELE BAI, RUI LI, AND DEXIN LI
School of Mechanical and Precision Instrument Engineering, Xi’an University of Technology, 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
Visual Simultaneous Localization and Mapping (VSLAM) system is considered to be a fundamental capability for autonomous mobile robots. However, most of the existing VSLAM algorithms adopt a strong scene rigidity assumption for analysis convenience, which ignored the influence of independently moving objects in the real environment on the accuracy of the SLAM system. Hence, this paper proposes MGC-VSLAM: Meshing-based and Geometric constraint VSLAM, a novel VSLAM algorithm for dynamic indoor environments, built on RGB-D mode of ORB-SLAM2, which relates to the problem of the ORB feature uniform distribution and the dynamic feature filtering. In detail, aiming at the problem of the over-uniform distribution of feature points extracted by the quadtree-based algorithm in ORB-SLAM2, a meshing-based feature uniform distribution algorithm is proposed. Meshes are divided at each layer of the image pyramid, and then a specific number of features in the meshes are reserved according to their Harris response value. In addition, aiming at the impact of features extracted from dynamic targets on the SLAM system, a dynamic feature filtering method is proposed. First, a stable matching relationship is established through a feature matching constraint method. Then a novel geometric constraint method is used to filter out the dynamic feature points in the scene. Only the remaining static features are reserved to achieve accurate camera pose estimation in dynamic environments. Experiments on the Oxford dataset and public TUM RGB-D dataset are conducted to evaluate the proposed approach. It revealed that the proposed MGC-VSLAM can effectively improve the positioning accuracy of ORB-SLAM2 in high-dynamic scenarios.

INDEX TERMS
Visual SLAM, ORB-SLAM2, dynamic indoor environment, uniform distribution, dynamic features filtering.

I. INTRODUCTION
Simultaneous Localization and Mapping (SLAM), as the core technology of intelligent mobile robots, refers to the fact that the robot simultaneously completes the positioning of the mobile robot itself and the map construction of the surrounding environment without any prior environmental information [1]–[2]. A SLAM system that only uses visual sensors to obtain external information is called a visual SLAM system (VSLAM). Some existing advanced VSLAM algorithms have achieved satisfactory results [3]–[6]. However, some problems in VSLAM have not been well solved until now [7]–[8]. For example, ORB features extracted by the standard ORB algorithm tend to concentrate on the strong texture regions. As a consequence, these features cannot reflect the whole image well, and the number of matched features will be significantly reduced when the concentrated regions of two adjacent frames are different [9]. This problem makes the SLAM system to be unstable and more seriously will cause tracking lost [10]. Besides, most existing algorithms use the external environment as a static assumption, ignoring the impact of dynamic targets on the accuracy of the SLAM algorithm in the real environment [11]–[12]. In the real environment, independent moving objects often appear in the scene. It will introduce errors into the visual odometry estimation, and moving objects will be recorded in the resulting map, which makes the built maps unsuitable for subsequent robot intelligent capture, navigation, and other
complex tasks [13]. Therefore, how to eliminate the negative impact of dynamic objects on the SLAM system is a critical challenge for VSLAM.

ORB-SLAM2 [3], [4] is one of the best VSLAM algorithms currently in use, which has high consistency positioning and map building accuracy. In ORB-SLAM2, ORB features [4] [14] are used for tracking. Given ORB features are prone to concentrate on the regions with strong texture. A quadtree-based method is adopted by ORB-SLAM2 to make features uniform distribution. It iteratively segments every layer of the image pyramid, extracting the feature with the largest Harris response value in child nodes [4]. However, features extracted by the quadtree-based algorithm are easy to over-uniform distribution, which keeps some weak quality features that not conducive to tracking, and the number of iterations affects its algorithm efficiency. A UR-SIFT algorithm is proposed by Sedaghat et al. [15], which can adaptively control the number of SIFT feature points in different regions. It is a better algorithm for uniformly distributing SIFT feature points. Paul and Pati [16] presented a modified UR-SIFT, which effectively generates robust enough, reliable, and uniformly distributed aligned feature points. Yao et al. [17] proposed an improved quadtree-based ORB algorithm. Different quadtree depths are set according to the expected number of features. However, this algorithm still has the problem of over-uniform distribution, and the matching state of the extracted features is not tested. So far, most of the feature uniform distribution algorithms proposed by scholars have high computational complexity, and the matching effect is not ideal.

Under all obstacles above, it remains the motivation for finding a novel VSLAM algorithm for dynamic indoor scenes. How to eliminate the ambiguity of motion caused by the camera’s motion and independent moving objects is the fundamental challenge of motion removal. The main problem in the SLAM is that camera is moving. The classic motion segmentation methods as the foreground and background move independently in the image, such as background subtraction and frame difference method, are useless [18]. Researchers have proposed many excellent motion removal methods. Berta Bescos et al. [19] add a dynamic target detection module on ORB-SLAM2, which can accurately identify and segment dynamic targets through the combination of multi-view geometry and deep learning. Similarly, Chao Yu et al. [20] filter the feature points extracted on the dynamic target through the epipolar constraint and Segnet network. However, the fundamental matrix used for the epipolar constraint has a large error, which affects the performance of the algorithm. Recently, deep learning networks as one of burgeoning methods which achieved good research results in the SLAM systems due to its strong scene adaptability. However, so far, high-precision deep learning segmentation networks require high-performance hardware to support real-time processing, which significantly increases application costs. Emrah [21]–[22] proposed a novel trigonometry-based kinematic positioning scheme, using a vertically suspended monocular camera and a novel geometric method to track the robot pose. The system performs well in the experimental environment, and it provides a new idea for the positioning of mobile robots for indoor environments. Wangsiririptak and Murray [23] avoid moving outliers through tracking known 3D dynamic objects. The work of Moo et al. [24] uses two single Gaussian models, which can effectively represent the foreground and the background. Sun et al. [11] extend the idea of using the intensity difference image to identify the boundary of dynamic objects. However, the huge amount of computation hinders real-time applicability. Kim et al. [12] propose to obtain the static parts of the scene by computing the difference between consecutive depths images projected over the same plane. Similarly, Zhao et al. [25] also use depth images to detect dynamic objects. However, these methods are prone to be affected by the uncertainty of depth images. More recently, Li and lee [26] use depth edge points with an associated weight indicating its probability of belong to a dynamic object.

To improve the stability and robustness of ORB-SLAM2 in the dynamic environments, a novel meshing-based uniform distribution approach is proposed, which solved the problems of the over-uniform distribution of features and the artificial FAST extraction threshold cannot suit the different environments. Moreover, a modified geometric constraint method is proposed to filter out the dynamic features, which can reduce the impact of the dynamic objects to the SLAM positng accuracy. The main content of this paper is arranged in the following section: Section 2 introduces the proposed improved SLAM algorithm system framework and the detailed process; Section 3 tests the proposed method experimentally; conclusions and future work are described in the last part of the paper.

II. METHOD
In this section, MGC-VSLAM will be introduced from three main aspects: system framework, feature uniform distribution algorithm, and dynamic feature point filtering algorithm.

A. SYSTEM OVERVIEW
The overview of the proposed MGC-VSLAM system is shown in Fig.1. It can be divided into three main threads: tracking, local mapping, and loop closing. First, a meshing-based feature uniform distribution algorithm is used to extract the stable ORB features on the current frame image. Then, the dynamic feature points extracted on the moving targets are filtered by a geometric constraint method. Only the static feature points are input into the tracking thread for pose estimation. The rest two threads keep the same as ORB-SLAM2.

B. IMPROVED ORB FEATURE POIT UNIFORM DISTRIBUTION ALGORITHM
In this section, a meshing-based feature uniform distribution method is introduced, which aims to solve the problem of the over-uniform distribution of features in ORB-SLAM2. As is
shown in Fig.2, it is a meshing model of the meshing-based algorithm. Detailed parameters of the meshing model will be illustrated in detail as follows.

1) MESH MODEL
First, the image pyramid is constructed to solve the problem of scale invariance of the ORB algorithm. The scaling ratio $S_i$ of each layer can be expressed as:

$$S_i = SF^i (i = 0, 1, \cdots, n - 1)$$ (1)

where $n$ is the total number of layers of the image pyramid, $SF$ is the scale factor of each layer of the image pyramid.

The $i$th layer of the image pyramid $I_i$ can be obtained by downsampling the $SF$ times of the image pyramid $I_{i-1}$.

Each layer of the image pyramid is divided into the meshes after obtaining the image pyramid. To make the feature points to be extracted distribute reasonably, the extracting number of each layer of the image pyramid is set as follows:

$$DesF_i = N \times \frac{1 - InvSF}{1 - InvSF^n} \times InvSF^i$$ (2)

where the total desired number of ORB features is denoted as $N$. $DesF_i$ is the number of feature points required for the $i$th layer. $InvSF$ represents the reciprocal of the scale factor, which can be expressed as $InvSF = 1/SF$.

The feature points in image edge are easily lost in two consecutive image frames, causing the matching to fail. Thus, a boundary is set on each layer of the pyramid image, and it is shown in Fig. 2 (a).

Hence, the width $ew_i$ and the height $eh_i$ of the feature point extraction area of the $i$th layer of the image pyramid can be expressed as:

$$ew_i = w_i - 2EdgeTH$$ (3)

$$eh_i = h_i - 2EdgeTH$$ (4)

where the width and height of the image of the $i$th layer of the image pyramid are denoted as $w_i$ and $h_i$, respectively. $EdgeTH$ is the boundary threshold.

Each mesh area should be as equal as possible to make the distribution of features more uniform. Therefore, the number of mesh rows and columns is set as equation (5).

$$\begin{cases} 
\text{Cols}_i = \sqrt{\frac{DesiredF_i}{ImRat_i \cdot t}} \\
\text{Rows}_i = ImRat_i \cdot \text{Cols}_i 
\end{cases}$$ (5)

where $t$ is the mesh division coefficient for controlling the number of the mesh. The aspect ratio of the $i$th layer of the image pyramid is $ImRat_i$. The full meshing process is shown in Fig. 3, where the meshing model is constructed to solve the problem of scale invariance of the ORB algorithm. The meshing model integrates the uniform distribution method and dynamic features filtering method into the tracking thread, by which only the static features are used to calculate and optimize the camera pose.
image pyramid is noted as $ImRat_i$, which can be expressed as $ImRat_i = eh_i/ew_i$.

Then the area of each mesh on the image pyramid can be finally determined, which can be expressed as:

$$\min X_{mn}(i) = n \times Gw_i - Th$$
$$\min Y_{mn}(i) = m \times Gh_i - Th$$
$$\max X_{mn}(i) = \min X_{mn} + Gw_i + Th$$
$$\max Y_{mn}(i) = \min Y_{mn} + Gh_i + Th$$

where $\min X_{mn}(i)$, $\min Y_{mn}(i)$, $\max X_{mn}(i)$ and $\max Y_{mn}(i)$ are the four boundary points of the mesh in row and column as shown in Fig2 (a). $Th$ is the mesh expansion coefficient, which can help better extract the features on the mesh edges. $Gw$ and $Gh$ are the actual feature extracting region of the mesh, which can be expressed as:

$$Gh_i = eh_i/Rows_i$$
$$Gw_i = ew_i/Cols_i$$

(10)

Then ORB features are extracted after each mesh is divided. The desired number of each mesh of the $i$ th layer of the image pyramid is set as equation (11) to make the distribution of feature points in each grid as uniform as possible.

$$cDesF_i = \frac{DesF_i}{Rows_i \cdot Cols_i}$$

2) ADAPTIVE THRESHOLD FAST FEATURE EXTRACTION IN THE MESH

The extraction threshold of FAST is usually set according to the engineering experience value in the standard ORB algorithm and ORB-SLAM2 algorithm, which cannot better meet the real environments. Such as, the high threshold cannot extract enough features in some image regions where the texture information is weak. Hence, a threshold that can adapt to image changes is set in this method, and the initial threshold $initTH$ is set as:

$$initTh = \frac{1}{\kappa \cdot n1} \sum_{x=1}^{m1} (I(x) - \kappa)^2$$

(12)

where $I(x)$ is the grey value of each pixel in the image, $\kappa$ is the average value of the image grey. The total number of the image pixel is represented as $n1$. FAST features are extracted with an initial threshold $initTH$ in the mesh. Reducing the threshold by a quarter when the number of the features extracted by the initial threshold is less than $cDesF_i$.

3) ORB FEATURE POINTS RETENTION STRATEGY IN THE MESH

First, ORB features are sorted by their Harris response value from large to small. Then the first $cDesF$ (the desired number of each mesh) items of Fast features in the mesh are reserved as the final ORB feature points. The detailed retention strategy flow is shown in Fig. 3.

In Figure3, $Kp$ is the final reserved features extracted by the proposed method. The total number of the pyramid layer is denoted as $n$, $m(i)$ represents the number of meshes in the $i$ th layer of the image pyramid. The number of features in the $j$ th mesh is denoted as $nf(j)$. $TemKp$ is the sequence to save temporary data.

### C. DYNAMIC FEATURE POINTS FILTERING ALGORITHM

How to filter out dynamic features is a huge challenge for feature-based VSLAM systems using in the real world. A novel geometric-based dynamic features filtering algorithm is introduced in this part. First, ORB feature points are extracted by the meshing-based algorithm for each frame, then the feature points between the adjacent frames are matched based on the BRIEF descriptors [27], which is accelerated through the bag-of-words model [28]. These methods are similar to the processing flow in the ORB-SLAM2 [3], [4] system, and will not be repeated here.

1) FEATURE POINT MATCHING ACCURACY CONSTRAINT METHOD (MAC)

The matched feature pairs using the BRIEF descriptors exist mismatches, which will lead to disorder of the dynamic feature point judgment method. Hamming distance is a quantification of the quality of feature matching. In view of this, the matched point pairs whose nearest Hamming distance $d_1$ is far greater than the next nearest Hamming distance $d_2$ is considered as a reliable match. In detail, a pair is considered as the correct match when they satisfy the relationship: $d_1/d_2 < \alpha$, where, $\alpha$ is the precision constraint coefficient, which controls the strictness of matching.

To better illustrate the matching state, a matching accuracy score is defined as:

$$q_a = \frac{d_1^2}{d_2}$$

(13)

The smaller the score is, the higher the probability the matching pair is the correct one, which will be used for the following dynamic feature detection.
2) FEATURE POINTS GEOMETRIC CONSTRAINT METHOD (GC)

The feature points will have a relatively stable matching relationship after they are filtered by the MAC method. However, the dynamic matched pairs are also retained. In view of this, an image feature point geometric constraint model is proposed.

There are three matched pairs in Fig. 4, for example, which construct two triangles $\Delta p_1 p_2 p_3$ and $\Delta q_1 q_2 q_3$ on the query image $I_1$ and the target image $I_2$, respectively. $I_1$ and $I_2$ are two adjacent frames, and its time interval is particularly small. Therefore, the projection distortion due to the camera pose change is very tiny. If there is no dynamic target in the scene, the corresponding edges between the two triangles should appear in a small interval, which can be expressed as:

$$d(p_i, p_j) - d(q_i, q_j) \in (d_{\min}, d_{\max})$$

To better describe this change of corresponding edges, a geometric constraint score function is defined as:

$$q_g(i, j) = \exp\left(\frac{d(p_i, p_j) - d(q_i, q_j)}{A(i, j)}\right)$$

where $d(a, b)$ denotes the Euclidean distance between $a$ and $b$ feature points. $A(i, j)$ represents the average distance between two features, which can be expressed as:

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

If there is a dynamic target that appears in the scene, for example, $q_1$ is extracted on this dynamic target, and $q_1$ moves to $q_1'$ on the target image $I_2$. As a consequence, the geometric constraint score $q_g$ calculated from formula (15) arises abnormally. However, there has a hard decision to identify which feature is dynamic due to the value of $q_g$ represents the common value of the two pairs of features.

In view of this, the following method called two-way scoring rule is adopted to identify the dynamic features. In detail, when an edge has an abnormal distance, adding one point to both the two feature points on the edge, that is, one’s abnormal score represents how many features judge it is a dynamic feature. Therefore, the abnormal score value of the real dynamic feature point will be unusually large when all features scored to each other. The mathematical expression of the abnormal score can be expressed as:

$$q_{ab}(i) = \sum_{j=0}^{M} s(i, j)$$

where $s(i, j)$ indicates the abnormal score increment, and it can be expressed as:

$$s(i, j) = \begin{cases} 1, & q_g(i, j) > \beta \cdot AS \\ 0, & \text{else} \end{cases}$$

where $\beta$ is the scale factor for the geometric score, which controls the strictness of the geometric constraint. $AS$ represents the average geometric score:

$$AS = \frac{\sum_{i,j=1}^{M} w_s^{i,j} q_g(i, j)}{M}$$

where $M$ is the number of matched pairs of feature points of the image. $w_s^{i,j}$ can be expressed as:

$$w_s^{i,j} = \begin{cases} 0, & q_g(i, j) + q_a(i, j) > \theta_h \\ 1, & \text{else} \end{cases}$$

where $\theta_h$ is the geometric score threshold, which means when the sum of one’s abnormal score and matching accuracy score larger than this threshold, then it is not used to calculate $AS$. Equation (20) can largely reduce the influence of the abnormal score of dynamic features on $AS$.

After all of the abnormal score $q_{ab}$ is calculated, an adaptive threshold method is used to filter out the dynamic features. For example, there have extracted 450 features in Fig.5 (b), and their abnormal scores are shown in Fig. 5(a). The red line in Fig. 5(a) represents the segmental threshold, which means if one’s abnormal score $q_{ab}$ is more than this threshold, then it is determined as the dynamic features. In detail, the segmental threshold is set as $\gamma M$, $M$ is the total number of the features; it is known that the abnormal score represents how many features judge one feature is dynamic. $\gamma$ is set as 60% in Fig.5, which means if 60% of features determine that one feature is abnormal then filter it. The filtered dynamic features are shown in Fig. 5(b) with red color. The more test of this algorithm is taken in the experimental part of this paper.
D. EXPERIMENTAL SETUP
Three experimental tests were performed to verify the effectiveness and robustness of the proposed algorithm, which is improved features uniform distribution algorithm, dynamic feature points filtering algorithm, and the improved VSLAM system performance test.

In the first experiment, Oxford datasets [29] are selected to test the uniformity and the matching accuracy. The public TUM dataset [30] is used to test the performance of the last two algorithms, which contains color and depth images with scale and rotation changes. The data is intended to evaluate the accuracy and robustness of the SLAM algorithm in indoor scenes with fast-moving dynamic objects. The real trajectory is obtained by a motion capture system, which consists of 8 high-speed (100HZ) cameras to ensure the reliability of the real trajectory. The dataset contains several typical types of camera self-motion. The camera has four types of self-motion: halfsphere, rpy, static, and xyz in the sitting and walking sequences, where, the halfsphere indicates that the camera moves along the hemisphere trajectory; the rpy indicates that the camera rotates along the roll-pitch-deflection axis; the Static indicates that the camera is basically fixed in place; the xyz indicates that the camera moves along the xyz axis.

All experiments were run on a computer with a 3.40 GHz Intel (R) i7-6700 CPU, 20GB of memory, and an Ubuntu 14.04 operating system. All tests are tested 5 times, and the average value is taken.

III. EXPERIMENTAL RESULTS
Three experiments were conducted. The purpose of the feature point uniform distribution method was to investigate the feature uniformity and matching accuracy. The dynamic feature filtering performance is tested in the second experiment. The last experiment SLAM test is designed to verify the positing accuracy of the improved VSLAM in dynamic environments.

A. IMPROVED ORB FEATURE POINT UNIFORM DISTRIBUTION ALGORITHM TEST

1) FEATURE POINTS UNIFORMITY TEST
The uniform degree [31] of the three algorithms was calculated in Table 1. The smaller the uniform degree value, the more uniform of the feature points. For the convenience of recording, in the table, the feature point uniform distribution algorithm in ORB-SLAM2 is noted as Qtree_ORB, and the proposed method is represented as Grid_ORB.

The experimental results show that the average uniform degree value of the Qtree_ORB is smaller than that of the Grid_ORB and ORB algorithm. It means that the feature points extracted by Qtree_ORB are more uniform than the other two algorithms.

2) FEATURE POINTS MATCHING PERFORMANCE TEST
The three algorithms were tested on the Oxford dataset in this section. SLAM problem usually relates to the scale change, Brightness change, view angle change, fuzzy change of the scene. Hence, the experimental results on these data will be analyzed as a focus. The matching results are shown in Fig. 6. The following is the detailed experimental data and analysis:
FIGURE 7. The results of the matching performance test. The three algorithms were tested on different scenes. The first column in the figure represents the test results of ORB, the second column denotes Qtree_ORB, and the third column is the matching results generated from Grid_ORB.

TABLE 2. Performance comparison of each algorithm under scale change.

| algorithm | Matching quantity | Correct matching quantity | CMR (%) |
|-----------|------------------|---------------------------|---------|
| ORB       | 91               | 86                        | 94.5    |
| Qtree_ORB | 79               | 76                        | 96.2    |
| Grid_ORB  | 129              | 128                       | 99.22   |

**a: SCALE CHANGE**

In terms of scale change, Fig.6 (a), (b) and (c) are the matching results of the three algorithms on bark data. The detailed experimental data is shown in Table 2.

It can be found from Table 2 that the Grid_ORB algorithm has a higher matching number than the other two algorithms. In terms of correct matching ratio (CMR), the Grid_ORB algorithm is about 5% better than that of the standard ORB algorithm. It can be seen from the query image of the Fig.6 (a) that the feature points extracted by the standard ORB algorithm are mostly concentrated on the bottom right corner with the strong texture. A large number of feature points on the aggregation region were not matched due to this region does not appear in the target image. The matching quantity of the Qtree_ORB is significantly impacted because it is reserved many weak feature points.

**b: BRIGHTNESS CHANGE**

The leuven data is used to verify the illumination robustness of the algorithms, and the matching results are shown in Fig.7 (d), (e), and (f). The detailed test data of each algorithm is shown in Table 3.

Table 3 shows Grid_ORB is superior to the other two algorithms in both the matching quantity and the CMR. It is about 3% higher than that of the other two algorithms in CMR. It can be seen from Fig. 6 (a) that the ORB algorithm cannot extract feature points in the weak and strong illumination area because it extracts feature points based on a fixed threshold. Conversely, Qtree_ORB and Grid_ORB based on adaptive thresholds, the extracted features are relatively uniform. However, the final matching quality is affected in Qtree_ORB because the feature extracted from Qtree_ORB tends to over-uniform, a large number of weak feature points are preserved, such as the feature points on the car glass and the ground.

**c: VIEW ANGLE CHANGE**

The data set graf and wall are used to test the impact of the view angle change on the algorithms. The experimental results are shown in Fig.7 (g)-(l), Table 4 and Table 5.

It can be found from Fig. 7 (g) that the feature points extracted by the standard ORB algorithm are mostly concentrated on the center of the image. Conversely, Qtree_ORB and Grid_ORB are relatively uniform. It can obtain from Table 4 that the
Grid_ORB algorithm has higher accuracy in the Graf and wall data and improves CMR by about 3% compared with the ORB algorithm and about 1.6% compared with Qtree_ORB algorithm.

d: FUZZY CHANGES

In terms of fuzzy changes, the matching results of the three algorithms on the bikes data are shown in Fig. 7 (m), (n) and (o), and the detailed test data of each algorithm is shown in Table 6.

As shown in Fig. 7 (m), feature points extracted by the ORB algorithm are mostly concentrated on the car, and the distribution of the features extracted from the other two algorithms is relatively uniform. In terms of the matching accuracy, the Grid_ORB algorithm is higher than ORB algorithm but lower than that of Qtree_ORB algorithm. The reason is that the distance of a few feature points extracted by the Grid_ORB algorithm is too close, which leads to the reduction of descriptor dissimilarity and ultimately leads to mismatching.

In terms of the efficiency of the algorithm, a comparison of the operation efficiency of each algorithm is shown in Table 7. The operation efficiency of the Qtree_ORB and Grid_ORB algorithm is slower than that of the ORB algorithm because they have added the feature point selecting module.

B. DYNAMIC FEATURE POINT FILTERING ALGORITHM TEST

In this section, the dynamic feature filtering algorithm is tested on the TUM dataset, and the results are shown in Fig. 8 and 9.

For convenience recording, the fr3_walking_xyx sequence in the TUM dataset is represented as w_zyx, and the TUM dataset fr3_walking_halfsphere dataset is denoted as w_half, the quadtree-based method is noted as “Q”, the meshing-based method is represented as “G”, and the feature geometric constraint method is noted as “GC”. The format “sequence/frame/method” represents used data, at which frame, and which method is used. The red box is used to indicate the regions where the feature points change significantly, and it does not represent all the changed regions.

The analysis will be performed from the scene including fast-moving objects and slow-moving objects.

1) THE ENVIRONMENT CONTAINS FAST-MOVING OBJECTS

The first column images in Fig. 8 are feature extraction results from Qtree_ORB. It can be seen that a large number of feature points are extracted in the moving object. Conversely, almost a few features are extracted by the Grid_ORB, which can be seen in the second column of Fig. 8. The reason for the different results of the two algorithms is that the motion blur is caused by fast-moving objects. In detail, the feature points descriptors extracted in the motion blur region have a small dissimilarity with others. These feature points have a low Harris response value, which belongs to weak features. However, Qtree_ORB needs to reserve at least one feature point in the child node. Hence, some weak features on the motion object are reserved. However, Grid_ORB will preferentially extract high-quality features, which makes it extract little feature on the moving objects. It is worth noting that this still works when the camera rotated.

In terms of dynamic features removal, the dynamic features extracted from Qtree_ORB are better filtered by the GC method, which can be seen from the images of the third column in Fig. 8. These results largely proved the effectiveness of the GC method. Conversely, Grid_ORB extracts little dynamic features, so the features filtering effect is not obvious.

2) THE ENVIRONMENT CONTAINS SLOW-MOVING OBJECTS

It can be seen from images of the first two columns in Fig. 9 that both algorithms have extracted dynamic features. The reason is that slow-moving objects have little impact on the image. That means the number of low-quantity features caused by moving objects is small in this scene, so Grid_ORB also extracts some dynamic features. In general, the dynamic features extracted from Grid_ORB are fewer than those of features extracted by Qtree_ORB. However, in Fig. 9 (d), Grid_ORB extracts more dynamic features than Qtree_ORB. In this scene, two people are chatting, the person on the left is stationary, and the person on the right is
slowly lifting the arm. In detail, as shown in the red box, the background of the arm of the moving target is a computer screen, and the grey difference of the hand edge changes significantly, so Grid_ORB extracts many robust features on the moving object.

In terms of dynamic filtering results, overall, the filtering effect of “G+GC” method is more obvious than that of the “Q+GC” method. The reason is that Qtree_ORB extracts a large number of dynamic features. These features have damaged the relationship between the features, causing the filtering failed of the dynamic features.

C. SLAM SYSTEM PERFORMANCE TEST

In this section, the proposed method is fused to the ORB-SLAM2 front-end as a preprocessing stage to filter out the dynamic features.

1) SLAM EXPERIMENT IN DYNAMIC ENVIRONMENTS

Quantitative evaluation was performed using absolute trajectory error (ATE) and relative pose error (RPE). Table 8-10 are the results of the quantitative test result. In the tables, MB-VSLAM represents the Grid_ORB
FIGURE 9. The feature extraction results when slow-moving objects appear in the scene.

TABLE 9. Experimental results of relative displacement trajectory error (RPE).

| Sequences | ORB-SLAM2 | Improved SLAM system |
|-----------|-----------|----------------------|
|           | rmse | std | rmse | std | rmse | std | rmse | std |
| w_xyz     | 0.4124 | 0.2684 | 0.0311 | 0.0159 | 92.46 | 94.08 | **0.0190** | **0.0093** |
| w_static  | 0.2162 | 0.1962 | 0.0172 | 0.0094 | 92.04 | 95.21 | **0.0087** | **0.0038** |
| w_rpy     | 0.4249 | 0.3166 | 0.4171 | 0.2968 | 1.84 | 6.25 | **0.1234** | **0.0940** |
| w_half    | 0.3550 | 0.2810 | 0.3450 | 0.2793 | 2.82 | 6.06 | **0.0251** | **0.0118** |
| s_half    | 0.0190 | 0.0122 | 0.0206 | 0.0136 | -8.29 | -11.58 | **0.0156** | **0.0077** |
| s_xyz     | 0.0112 | 0.0056 | 0.0121 | 0.0060 | -8.11 | -7.12 | **0.0109** | **0.0055** |

Method is fused to the ORB-SLAM2. MGC-VSLAM denotes both Grid_ORB and GC method are all fused to ORB-SLAM2. IMPROVEMENT represents the percentage improvement of the test algorithm over the standard algorithm.

RMSE in the table is highlighted as it is susceptible to large or occasional errors [32]. Therefore, the robustness of the SLAM system can be better represented by RMSE. STD is also highlighted because it represents the stability of the system. In the sitting sequence, the dynamic targets are the local limb movement of two people, which belongs to a low dynamic motion sequence. In the tables, the low dynamic motion sequence is denoted as “∗”.

The test results of absolute trajectory errors are shown in Table 8, and the running trajectory of each algorithm on the two test sequences is shown in Fig. 10 and 11.

Table 8 shows the performance of the improved system is significantly improved compared to the original system in the
TABLE 10. Experimental results of Relative rotational trajectory error (RPE).

| Sequences | ORB-SLAM2 | MB-VSLAM | IMPROVEMENT | MGC-VSLAM | IMPROVEMENT |
|-----------|-----------|----------|-------------|-----------|-------------|
|           | rmse  | std | rmse  | std | rmse  | std | rmse  | std | rmse  | std |
| w_xyz     | 7.7432 | 4.9895 | 0.6958 | 0.4278 | 91.01 | 91.43 | 0.6011 | 0.3758 | 92.24 | 92.47 |
| w_static  | 3.8958 | 3.5095 | 0.3840 | 0.1857 | 90.14 | 94.71 | 0.2483 | 0.1066 | 93.63 | 96.96 |
| w_rpy     | 8.0802 | 5.9499 | 7.1397 | 4.9022 | 11.64 | 17.61 | 2.2798 | 1.6936 | 71.79 | 71.54 |
| w_half    | 7.3744 | 5.7558 | 7.1503 | 5.6352 | 3.04 | 2.10 | 0.7539 | 0.3744 | 89.78 | 93.49 |
| s_half    | 0.5711 | 0.2598 | 0.5413 | 0.2661 | 5.21 | 2.41 | 0.5825 | 0.2804 | -2.01 | -7.92 |
| s_xyz     | 0.4877 | 0.2685 | 0.5807 | 0.2945 | -19.07 | -9.69 | 0.4763 | 0.2587 | 2.33 | 3.63 |

high dynamic test sequence. The reason is that the improved system can effectively filter out the features extracted on the dynamic targets, avoiding the influence of the dynamic features on the SLAM positioning estimation. It can be seen from the test data that the positioning accuracy of ORB-SLAM2 can be effectively improved simply integrates the Grid_ORB method. MGC-VSLAM reduces the average RMSE of ORB-SLAM2 on highly dynamic sequences by about 93%. This proves that the improved system can effectively improve the performance of ORB-SLAM2 in high dynamic scenarios.

However, the accuracy of ORB-SLAM2 is reduced by MB-VSLAM in the low dynamic sequences. The reason is that Grid_ORB extracted a large number of dynamic features, which introduces errors into the SLAM system. But this defect is made up by the GC method. More high-quality feature points on the background are extracted by the Grid_ORB method, which provides a stable matching relationship between image frames makes it can better to do geometric constraints. ORB-SLAM2 system, can well filter out the dynamic feature points extracted from low dynamic sequences because of the local map mechanism, is already has high accuracy, which improvement room is very small. Hence, MGC-VSLAM system has comparable performance to ORB-SLAM2 in a low dynamic sequences.

It is worth noting that ORB-SLAM2 performed unstable in the w_rpy sequence. This sequence scene contains fast-moving targets, and the camera’s rotation and translation. Through analysis, the reason is that the w_rpy sequence contains a large number of images with motion blur and weak texture information. Qtree_ORB retains too many weak and dynamic feature points, resulting in unstable feature matching, and then causing system disruption. Conversely, the improved system is relatively stable.

The performance of the visual odometer is shown in tables 9 and 10. The performance of the visual odometer can be effectively improved by the proposed method when the fast-moving objects appear in the scene. Unfortunately, MGC-SLAM was found to degrade the performance of ORB-SLAM2 in the sitting sequence.

2) COMPARISONS WITH THE STATE-OF-THE-ART SLAM SYSTEMS
MGC-VSLAM is compared with state-of-the-art SLAM systems in a dynamic environment. DynaSLAM [19], DS-SLAM [20], Detect-SLAM [33] and the improved system proposed by Lin et al[34] are adopted for comparisons. They are all excellent VSLAM proposed in recent years. The above systems are all built upon ORB-SLAM2, and they adopt RMSE of ATE as a quantitative metric. However, we notice that the results of ORB-SLAM2 in the same sequence are different between these papers and ours. This may be caused by the difference in the experiment condition. So the relative accuracy improvement of each system to ORB-SLAM2 is adopted as the evaluation metric. The relative accuracy improvement is shown in Table 11. In terms of relative accuracy improvement, MGC-VSLAM is only lower than Lin’s system in w_static sequence and DynaSLAM in w_rpy sequence. This is largely due to the proposed Grid_ORB and GC method can effectively filter the features extracted on dynamic objects.

The reason why Lin’s system is better than MGC-VSLAM in w_static sequence is that the proposed GC method is not sensitive to slow-moving objects; some reserved dynamic features may impact the accuracy of the MGC-VSLAM. However, Lin’s system adopts a semantic method to detect a priori dynamic objects to avoid the impact of some dynamic objects on the system, which may improve the accuracy of its system in the w_Static sequence; in the w_rpy sequence, it contains many numbers of frames with a large area of motion blur. Dynaslam has a higher accuracy may due to its deep learning method, which has better adaptability to the scene. However, many weak features extracted by Grid_ORB impacts the decision of the GC method, causing the lower accuracy of MGC-VSLAM.

D. DISCUSSION
MGC-VSLAM is designed to improve the robustness of ORB-SLAM2 in the dynamic environments, which improved the feature extracting method and added a dynamic feature point filtering module in ORB-SLAM2.
According to these above tests, the feature points extracted by the proposed Grid_ORB are more uniform than that of extracted from standard ORB\cite{9} and does not exist the over-uniform distribution problem such as in Qtree_ORB \cite{3}, \cite{4}. Moreover, the extraction threshold of FAST is set according to the engineering experience value in standard ORB and ORB-SLAM2 algorithm, which cannot better meet the real environments. Grid_ORB adopts an adaptive threshold, which can better suit to the different environments; In terms of efficiency, the running efficiency of the Qtree_ORB is severely affected because each time the child node is divided, it is necessary to determine which child node the feature point in the parent node belongs to, and the more serious problem is that the number of iterations is too large. Grid_ORB extracts feature points in each mesh, the computational efficiency is faster than that of the Qtree_ORB. The proposed dynamic filtering method called GC can effectively filter the features extracted from Grid_ORB and Qtree_ORB in a high dynamic scene, and it is also useful when the camera with rotation and translation. But GC is not sensitive to slow-moving dynamic, which may reserve some dynamic features, impacting the accuracy of the SLAM. The biggest reason is that the projection distortion larger than the transformation caused by the moving object.

MGC-VSLAM was tested in public TUM dataset to validate the positing performance, which reduces the average RMSE of ATE in ORB-SLAM2 \cite{3}, \cite{4} on high dynamic sequences by approximately 93%. Unfortunately, the stability is reduced by MGC-VSLAM when slow-moving objects appear in the scene. Moreover, the state-of-art SLAM system: DS-SLAM \cite{20}, Dynaslam \cite{19}, Detect-SLAM \cite{31}, and Lin’s system \cite{32} are compared with MGC-VSLAM. The results show that the relative accuracy improvement
of MGC-VSLAM is only lower than Lin’s system in the w_static sequence and lower than Dynaslam in the w_rpy sequence. In the low dynamic sequence, the performance of MGC-VSLAM is better than all compared SLAM systems. It is worth noting that all four systems compared with MGC-VSLAM are based on deep learning methods, which require high-performance graphics cards for support. Conversely, MGC-VSLAM system does not require, so it has lower operating costs.

There are still more ongoing works on MGC-VSLAM. It may be improved in two aspects. First, the dynamic feature point detection in MGC-VSLAM only uses two consecutive frames. Further improvement may be achieved by using more frames, which may provide more redundant information to filter dynamic feature points. Second, MGC-VSLAM currently adopts a hard decision way to determine whether slow-moving objects appear in the scene. We are considering using deep learning methods to identify a priori dynamic objects, and then filter the features on these objects.

IV. CONCLUSION

In this paper, a novel meshing-based and Geometric constraint VSLAM, i.e., MGC-VSLAM, is proposed to overcome the degeneration of the SLAM system in high dynamic environments. MGC-VSLAM uses a new approach called Grid_ORB to extract more uniform and stable ORB features. Then a dynamic feature point filtering method GC was added to the system to filter out dynamic feature points. The following conclusions are obtained through experimental verification:

1) The proposed Grid_ORB can effectively solve the problem of the over-uniform of features extracted in ORB-SLAM2. Moreover, the average matching accuracy and efficiency are higher than those of Qtree_ORB and ORB.

2) The proposed GC method can better filter out the dynamic features in the high dynamic scenes. However, GC is not sensitive to features extracted on the low dynamic objects.

3) In the high dynamic test sequences, MGC-VSLAM decreases about 93% of the average RMSE of ATE compared with ORB-SLAM2. However, the stability is reduced in the low dynamic sequences. In addition, the comparison with the four state-of-the-art SLAM systems in dynamic environments shows that MGC-VSLAM achieves the highest relative RMSE reduction with respect to ORB-SLAM2.

REFERENCES

[1] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, “Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios,” IEEE Robot. Autom. Lett., vol. 3, no. 2, pp. 994–1001, Apr. 2018.
S. Yang et al.: MGC-VSLAM: Meshing-Based and Geometric Constraint VSLAM for Dynamic Indoor Environments

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

[3] R. Mur-Artal, 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.

[4] 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.

[5] T. Qin, P. Li, and S. Shen, “VINS-mono: A robust and versatile monocular visual-inertial state estimator,” IEEE Trans. Robot., vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

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

[7] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” IEEE Trans. Robot., vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[8] T. Taketomi, H. Uchiyama, and S. Ikeda, “Visual SLAM algorithms: A survey from 2010 to 2016,” IPSJ Trans. Comput. Syst. Appl., vol. 9, no. 1, p. 16, Dec. 2017.

[9] 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.

[10] Y. Zhang, T. Jiang, and G. W. Jiang, “Uniform distributed subpixel ORB feature extraction method for high-precision SLAM,” Opt. Precis. Eng., vol. 26, no. 10, pp. 2575–2583, 2018.

[11] 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.

[12] 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.

[13] M. R. U. Saputra, A. Markham, and N. Trigoni, “Visual SLAM and structure from motion in dynamic environments,” ACM Comput. Surveys, vol. 51, no. 2, pp. 1–36, Jun. 2018.

[14] 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.

[15] A. Sedaghat, M. Mokhtarrzade, and H. Ebadi, “Uniform robust scale-invariant feature matching for optical remote sensing images,” IEEE Trans. Geosci. Remote Sens., vol. 49, no. 11, pp. 4516–4527, Nov. 2011.

[16] 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.

[17] J. Yao, P. Zhang, Y. Wang, Z. Luo, and X. Ren, “An adaptive uniform distribution ORB based on improved quadtree,” IEEE Access, vol. 7, pp. 143471–143478, 2019.

[18] M. Yazdi and T. Bouwmans, “New trends on moving object detection in video images captured by a moving camera: A survey,” Comput. Sci. Rev., vol. 28, pp. 157–177, May 2018.

[19] 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.

[20] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, “DS-SLAM: A semantic visual SLAM towards dynamic environments,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 1168–1174.

[21] E. Dönmez, A. F. Kocamaz, and M. Dirik, “A vision-based real-time mobile robot controller design based on Gaussian function for indoor environment,” Arabian J. Sci. Eng., vol. 43, no. 12, pp. 7127–7142, Dec. 2018.

[22] E. Dönmez and A. F. Kocamaz, “Design of mobile robot control infrastructure based on decision trees and adaptive potential area methods,” Iranian J. Sci. Technol., Trans. Electr. Eng., vol. 44, no. 1, pp. 431–448, Mar. 2020.

[23] S. Wang, Y. Zhang, and D. W. Murray, “Avoiding moving outliers in visual SLAM by tracking moving objects,” in Proc. IEEE Int. Conf. Robot. Autom., May 2009, pp. 375–380.

[24] K. M. Yi, K. Yun, S. W. Kim, H. J. Chang, and J. Y. Choi, “Detection of moving objects with non-stationary cameras in 5.8ms: Bringing motion detection to your mobile device,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, Jun. 2013, pp. 27–34.

[25] 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.

[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] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “BRIEF: Binary robust independent elementary features,” in Proc. Eur. Conf. Comput. Vis., vol. 63, no. 14, 2010, pp. 778–792.

[28] T. Joachims, “Text categorization with support vector machines: Learning with many relevant features,” in Proc. Eur. Conf. Mach. Learn., Berlin, Germany: Springer, Heidelberg, 1998, pp. 137–142.

[29] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[30] 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., Oct. 2012, pp. 573–580.

[31] C. Yi and T. Xiaoyu, “Research on visual SLAM method of mobile robot based on improved ORB algorithm,” Appl. Electron. Technique, vol. 45, no. 1, pp. 10–13, 2019.

[32] C. Kerl, J. Sturm, and D. Cremers, “Robust odometry estimation for RGB-D cameras,” in Proc. IEEE Int. Conf. Robot. Autom., May 2013, pp. 3748–3754.

[33] 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. Comput. Vis. (WACV), Mar. 2018, pp. 1001–1010.

[34] L. Cu and C. Ma, “SOF-SLAM: A semantic visual SLAM for dynamic environments,” IEEE Access, vol. 7, pp. 166528–166539, 2019.

SHIQUANG YANG was born in Baiyin, Gansu, China, in 1973. He received the Ph.D. degree in mechanical engineering from the Xi’an University of Technology of China, Xi’an, China, in 2010. From 2005 to 2018, he was with the Xi’an University of Technology of China. Since 2009, he has been an Associate Professor with the School of Mechanical and Precision Instrument Engineering, Xi’an University of Technology. From 2011 to 2018, he conducted the master research with the School of Mechanical and Precision Instrument Engineering, Xi’an University of Technology. His current research interests include the intelligent robot control, image recognition, behavior detection, and recognition.

GUOHAO FAN was born in Baotai, Shaanxi, China, in 1994. He received the B.S. degree in mechanical and electronic engineering from the Shaanxi University of Technology of China. He is currently pursuing the M.S. degree in mechanical and electronic engineering with the Xi’an University of Technology. His research interests include visual SLAM and image processing.
LELE BAI was born in Weinan, Shaanxi, China, in 1996. He received the B.S. degree in mechanical engineering from the Xi’an University of Technology of China, in 2018, where he is currently pursuing the M.S. degree. His research interests include visual SLAM and image processing.

DEXIN LI was born in Yantai, Shandong, China, in 1965. He received the Ph.D. degree in mechanics from the Xi’an Jiaotong University of China, Xi’an, China, in 2003. From 1992 to 2019, he was with the Xi’an University of Technology of China. 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 interest includes mechanical computer-aided design and manufacturing.

RUI LI received the M.S. and Ph.D. degrees in mechanical engineering from Xi’an Jiaotong University, Xi’an, China, in 2013 and 2019, respectively. She is currently working with the Xi’an University of Technology. Her research interests include intelligent robotics, brain computer interface, and brain controlled prosthesis.

* * *