Multimedia Tools and Applications (2021) 80:31729–31751
https://doi.org/10.1007/s11042-021-11168-5

GMSK-SLAM: a new RGB-D SLAM method with dynamic areas detection towards dynamic environments

Hongyu Wei [1,2] · Tao Zhang [1,2] · Liang Zhang [1,2]

Received: 20 June 2020 / Revised: 8 May 2021 / Accepted: 22 June 2021 /
Published online: 19 July 2021

Abstract
As a research hotspot in the field of robotics, Simultaneous localization and mapping (SLAM) has made great progress in recent years, but few SLAM algorithms take dynamic or movable targets in the scene into account. In this paper, a robust new RGB-D SLAM method with dynamic area detection towards dynamic environments named GMSK-SLAM is proposed. Most of the existing related papers use the method of directly eliminating the whole dynamic targets. Although rejecting dynamic objects can increase the accuracy of robot positioning to a certain extent, this type of algorithm will result in the reduction of the number of available feature points in the image. The lack of sufficient feature points will seriously affect the subsequent precision of positioning and mapping for feature-based SLAM. The proposed GMSK-SLAM method innovatively combines Grid-based Motion Statistics (GMS) feature points matching method with K-means cluster algorithm to distinguish dynamic areas from the images and retain static information from dynamic environments, which can effectively increase the number of reliable feature points and keep more environment features. This method can achieve a highly improvements on localization accuracy in dynamic environments. Finally, sufficient experiments were conducted on the public TUM RGB-D dataset. Compared with ORB-SLAM2 and the RGB-D SLAM, our system, respectively, got 97.3% and 90.2% improvements in dynamic environments localization evaluated by root-mean-square error. The empirical results show that the proposed algorithm can eliminate the influence of the dynamic objects effectively and achieve a comparable or better performance than state-of-the-art methods.

Keywords Dynamic environments · SLAM · GMS algorithm · Moving areas detection

✉ Tao Zhang
zhangtao22@seu.edu.cn

[1] School of Instrument Science & Engineering, Southeast University, Nanjing 210096, China

[2] Key Laboratory of Micro-Inertial Instrument & Advanced Navigation Technology, Ministry of Education, Nanjing 210096, China

1 Introduction

In recent years, Simultaneous Localization and Mapping (SLAM) technology has become a fundamental prerequisite in many applications [7, 38], such as robots, driverless cars, VR, 3D reconstruction, and so on. Because of its ability to conduct navigation and perception simultaneously in an unknown environment, SLAM has attracted the attention of many scholars and gradually become a research hotspot over the past decades [7]. The framework of the modern visual SLAM system is quite mature, which consists of several essential parts: feature extraction front-end, state estimation back-end, loop closure detection, and so forth [40]. Visual SLAM, the main sensor of which is the camera, can be parted to monocular slam, RGB-D slam, and stereo slam based on camera types. The monocular camera has practical advantages on size, power, and cost, but also has several disadvantages, such as scale ambiguity, complex initialization, and weak system robustness. However, the absolute scale of the system can be obtained by using stereo or RGB-D cameras, and the stability of the system can be enhanced at the same time [19]. In order to reduce the error caused by the scale estimation, the RGB-D camera with depth information or the stereo camera that can calculate the depth are more popular in experiments and scientific research.

Most indoor SLAM methods are based on the assumption that the environment is static. In other words, the geometric distribution of objects in the scene is assumed to be stationary in the process. However, this is unrealistic in real life. The environment in which people live is dynamic, and there will be many variable factors in the scene such as lighting, dynamic targets, occlusion, etc. Aiming at these problems, many scholars at home and abroad have also carried out rich and detailed research. In order to improve the environmental adaptability of the positioning system, this paper mainly focuses on the research of dynamic targets in the scenes. Most state-of-the-art dynamic environment positioning methods using vision-only are neural network-based and image-based. With the development of artificial intelligence, more and more people are introducing neural networks into SLAM. These methods designed various types of convolutional neural networks to try to segment the target information in the image. The camera pose estimation accuracy is improved by removing the target. The more classic and efficient target detection algorithm is the YOLOv3 [28]. Many papers use YOLOv3 algorithm for target detection, and divide them into dynamic targets or static targets based on the types of targets. Although the CNN-based method can accurately segment the target in the scene, due to the limitation of the training dataset, its detection accuracy will be constrained by the test dataset, so the purpose of removing dynamic targets cannot be well achieved. Image-based dynamic environment detection algorithms are more inclined to traditional image processing algorithms, such as [38]. This type of algorithm also has a good performance in dynamic target segmentation, which improves the stability and adaptability of the positioning algorithm in the environment to a certain extent. Although these algorithms have improved the camera's positioning accuracy in a dynamic environment, they have poor environment adaptability. These methods tend to classify all objects in a class of tags, such as people and cars, as dynamic, and then remove these areas directly from the images, which will not only reduce the system's ability to understand the environment, but also lose most of the feature points of the image. This is a fatal flaw in feature-based SLAM algorithms. In this paper, the GMSK-SLAM system is proposed, which not only detects dynamic regions in the images rather than the dynamic targets but also improves the localization accuracy in a dynamic environment. In order to eliminate the influence of dynamic objects on posture estimation results, the usual methods discard all feature points on dynamic objects, which is a rough way

for the whole system. In this case, even if the person in the scene does not make any posture changes,these methods still detect this person as dynamic and eliminate all feature points on this person. However, in our method, this person will be detected as a static object, the feature points will be kept and used to estimate the camera pose. The overview of this paper is shown in Fig. 1. The main contributions of this paper are identified as followed:

1. A complete dynamic SLAM system in changing environments is proposed based on ORB-SLAM2 [25], which could reduce the influence of dynamic objects on camera pose estimation. The effectiveness of the system is evaluated on TUM RGB-D dataset [35]. The results indicate that GMSK-SLAM outperforms ORB-SLAM2 significantly regarding accuracy and robustness in dynamic environments.
2. Put a dynamic area detection in an independent thread, which combining the GMS (Grid-based Motion Statistics)feature points detection algorithm [4] with k-means cluster algorithm [22] to filter out a dynamic portion of the scene, like walking people. Different from other methods, this paper innovatively proposes the concept of dynamic area, and focuses on dynamic regions rather than dynamic objects. The application of dynamic regions eliminates the limitation of the algorithm and is applicable to both rigid targets and deformable targets. Thus, the performance of the localization module is improved in respect of robustness and accuracy in dynamic scenarios.
3. GMSK-SLAM creates a separate thread to detect dynamic areas, which greatly improves the operating efficiency and robustness of the system. The proposed system could run the tracking thread and the dynamic area detection thread in parallel, thus allowing our system to read read-time performance.

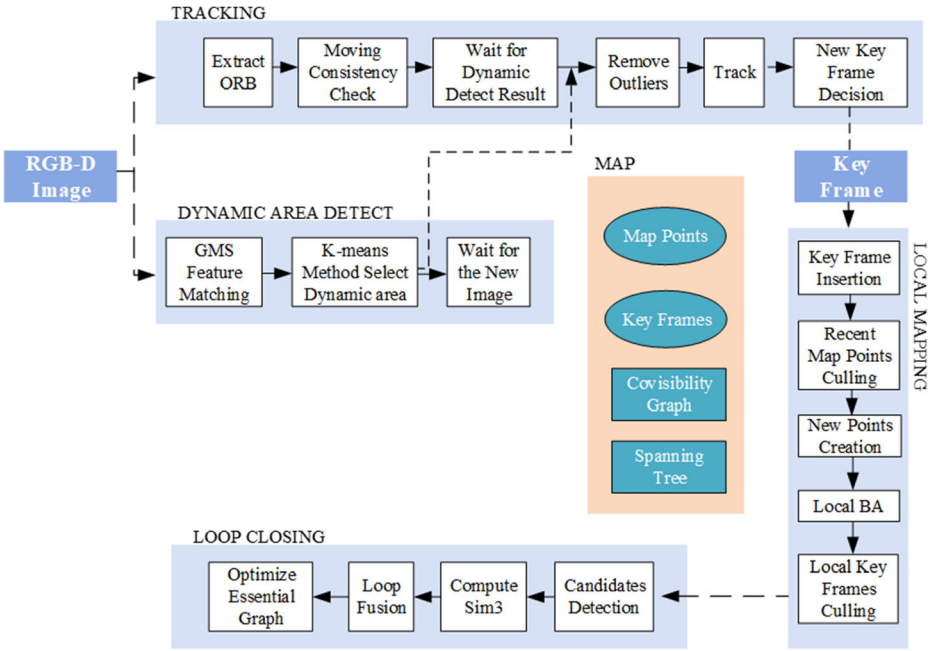

Fig. 1 The overview of GMSK-SLAM

The rest of this paper is organized as follows. First, visual SLAM and SLAM in dynamic environment related work are discussed in Section 2. The main work is described and demonstrated in Section 3 in detail, and a series of evaluation results and dataset test are presented in Section 4. Finally, the conclusion is made in Section 5.

2 Related work

2.1 Visual SLAM

Simultaneous Localization and Mapping (SLAM) [22] has been developed for about thirty years and has become a crucial technology in the field of robotics, automation, and computer vision. Recent years, researchers have proposed many effective open-source SLAM methods, such as Catragrapher [39], Hector SLAM [34], Gmapping [14], Karto SLAM [16], which are based on laser sensor SLAM and monoSLAM [6], ORB-SLAM [26], LSD-SLAM [9], RGB-D SLAM [8],DSO [10], and SVO SLAM [12], which are based on visual slam.

Especially for visual SLAM, visual SLAM has drawn the attention of researchers because of its low cost and small size. In 2007, Davison A. J. proposed Mono-SLAM [6] for the first time to implement a monocular real-time slam system. This work extends the range of robotic systems in which SLAM can be usefully applied but also opened up new areas. Subsequently, the PTAM [15] (Parallel Tracking and Mapping) is proposed by Klein, which firstly divided front-and-end in SLAM into two threads: one thread is mainly responsible for estimating camera posture, another mainly for restoring 3D information of feature points, which called mapping. At the same time, during the mapping and tracking processes, PTAM adopted keyframes strategy, used keyframes to estimate camera pose, which improved the accuracy of mapping and tracking. In 2014, the LSD-SLAM [9] proposed by Engel firstly achieved semi-dense scenes reconstruction in ordinary CPU, and operated directly on image pixels. Meanwhile, Forster proposed a fast Semi-direct monocular Visual Odometry (SVO) [12], which combined the feature point and direct tracking optical flow methods. Later, frameworks such as DSO [10] and VINS-Mono [37] have been proposed on after another. In two algorithms of extracting feature points of SLAM, the feature method also has its advantages. On the one hand, feature extraction and matching can ensure the accuracy of pose estimation in the SLAM tracking process. On the other hand, the feature method can extract more effective information from visual images, such as semantics, object recognition, feature localization, etc. In general, feature-based SLAM is more respected by scholars. There are also some excellent SLAM methods based on feature tracking. ORB-SLAM uses ORB (Oriented FAST and rotated BRIEF [5]) feature points [31] to match two consecutive frames and calculate camera pose, back-end use bundle adjustment method to optimize pose and map points, loop closing module transfer images to words by bag of words, and use the optimization method to make global optimization. The ORB-SLAM2 [25] framework is the successor of the classic SLAM thread model. It can work with monocular, stereo, and RGB-D three types of sensors. It follows most of the ideas in PTAM and adds initialization module, loop closing detection module, loop closing rectification module, and relocation module. However, these methods are unable to distinguish the features in static and dynamic objects, which leads to the deterioration of SLAM systems because of erroneous data association and faults in motion estimation. Therefore, it is necessary to make further exploration for it still has many shortcomings in dealing with dynamic environment problems.

2.2 SLAM in dynamic environments

A basic assumption of most current SLAM methods is that the environment is static. Nevertheless, active objects like humans and cars, always exist in many real-world scenes. Therefore, these methods originally to perform SLAM in a static environment cannot work well in complex dynamic environments. To solve this problem, we need to recognize moving areas or objects from the environment and exclude these objects or areas before pose estimation.

For dynamic environment problems, in recent years, researchers have proposed different methods to reduce camera pose errors. There are two main categories of methods, one is based on various convolutional neural networks, such as FCN (Full convolutional network) [20], SegNet [1] and YOLO [29], etc., the other is based on the traditional image process, such as optical flow technique. Optical flow is generated if movements exist in the image, so static background and moving target could be distinguished by computing the inconsistency of optical flow. Seungwon Oh [27] proposed the Dynamic Extended Kalman Filter SLAM based on the independence of the dynamic landmarks. Jia-Ning Li [18] proposed a scalable dense frame-to-frame model SLAM system based on KinectFusion algorithm. Fang [11] used optimum-estimation and uniform sampling methods to detect dynamic objects. In 2017, Sun [36] proposed an improving RGB-D SLAM method, which acted as a pre-processing stage to filter out data that were associated with moving objects. In 2018, Bahraini [2] proposed an approach to segment and track multiple moving objects by using Multi Level-RANSAC. Kajal Sharma [33] proposed a novel approach to mapping and localization which is based on detecting stable and invariant landmarks in consecutive RGB-D frames of the robot dataset. Muhamad [32] surveyed the problems of visual SLAM and Structure from Motion (SfM) In dynamic environments. Zhang [41] integrated a deep CNN model to improve the accuracy of terrain segmentation and make it more robust against wild environments. In 2019, Sang Jun Lee [17] presented a sampling-based method that improved the speed and accuracy of the existing Visual Bag of Words models. They first proposed sampling of image features considering their density to speed up the quantization. In the same year, Runzhi Wang [38] proposed a new RGB-D SLAM with moving object detection for dynamic indoor scenes, which is based on mathematical models and geometric constraints and can be incorporated into the SLAM process as a data filtering process. Mu [23] proposed a novel features Selection algorithm in a dynamic environment, which is merged into MSCKF based on trifocal tensor geometry.

3 Methodology

This section mainly describes the details of the proposed method. Firstly, the architecture chart of the framework of GMSK-SLAM is presented in Fig. 1. Secondly, we briefly explain the GMS feature matching algorithm used in this paper. Subsequently, the k-means clustering algorithm that is used for distinguishing dynamic areas from the image is introduced. Finally, the dynamic area rejection method is demonstrated, which combines the area detection and the moving consistency check to filter out dynamic feature points.

3.1 Framework of GMSK-SLAM

In practice, the two critical factors to evaluate autonomous robots are accurate pose estimation and its reliability in harsh environments. In recent years, ORB-SLAM2 is very popular in

visual-only SLAM and has excellent performance. However, ORB-SLAM2 is still facing some problems, such as it not expressive enough in dynamic environment. Hence, we propose an enhanced robust system based on ORB-SLAM2, which can not only enhance environment adaptability but also improve the positioning accuracy. GMSK-SLAM combines the GMS feature matching algorithm with the k-means cluster algorithm, which can perform pose tracking on dynamic scenes and with the better positioning accuracy.

As shown in Fig. 1, four threads run in parallel in GMSK-SLAM: tracking, dynamic area detection, local mapping, and loop closing. RGB images and Depth images are captured by a RGB-D camera in the scenes. The raw RGB images are processed in the tracking thread and dynamic area detection thread simultaneously. The tracking thread extracts ORB feature points by ORB feature matching, then checks moving consistency of feature points and saves the potential outliers. Then the tracking thread waits for the image that has been detected by dynamic area detection thread. In the dynamic area detection thread, GMSK-SLAM uses the GMS feature matching algorithm to match feature points and the sliding windows model to count filled-matching points, then adopts the k-means cluster algorithm to distinguish the dynamic area and static area. After the detection result arrives, the ORB feature points outliers located in moving objects will be discarded. Finally, the camera pose matrix is calculated by matching the rest of the stable feature points.

3.2 Feature matching by GMS

In the dynamic area detection thread, the system uses the GMS feature matching algorithm to enhance matching results which is an efficient and real-time feature matching algorithm. The limitation of the classical feature matching algorithms (SIFT [21], SURF [3], Hamming Distance, Brute Force Matcher, FANN [24], ORB[13, 30]) is that the algorithm with strong robustness has low matching speed, while the algorithm with high matching speed has poor robustness. However, compared to traditional algorithms, the advantage of GMS is that the algorithm has better performance in terms of time and accuracy.

GMS algorithm is an evaluation method based on matching to the support of neighborhood feature points. It mainly filters the feature matching after ORB feature extraction and BF matching. Figure 2 shows the matching schematic diagram of the two images. For image pairs $\{I_a, I_b\}$, they have {N, M} features respectively, and $\chi = \{x_1, x_2 \cdots, x_i, \cdots, x_j, \cdots, x_N\}$ is the

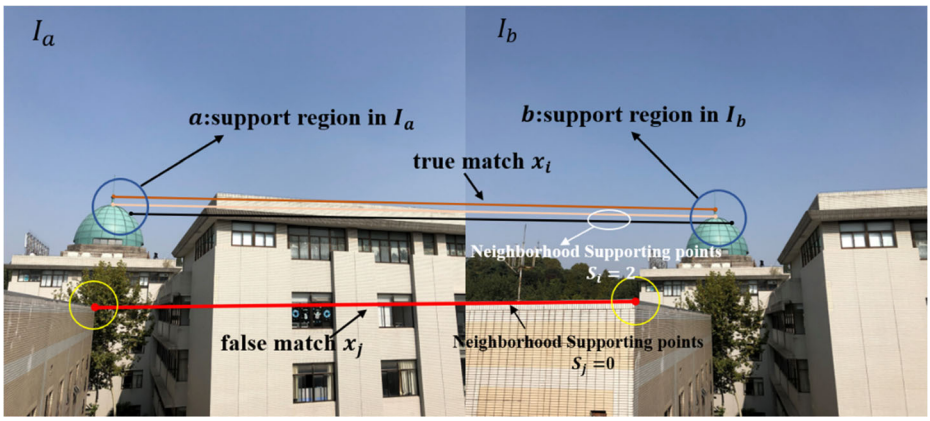

Fig. 2 Schematic diagram of GMS neighborhood feature point support principle

matching set from I_a to I_b, where $x_i = (a_i, b_i)$. GMS method separates χ into the sets of true and false matches by analyzing the local support of each match. Let region pairs $\{a, b\}$ which each with $\{n, m\}$ additional features respectively are respective regions of $\{I_a, I_b\}$, and f_a with correct matching probability t denotes one of n supporting features in regions a. GMS method assumes that there are M possible locations of f_a's nearest neighbor matching lying. Then we can get:

$$p\left(f_a^b | f_a^f\right) = \beta m/M \quad \beta \epsilon (0,1) \tag{1}$$

Where f_a^b denotes that f_a 's nearest neighbor is a feature in region b, f_a^b denotes that f_a matches wrongly, and β is a weighting factor that is added to accommodate a violation of the assumption by having a repeating structure. Let P_t, P_f denote the probability of f_a^t and f_a^f respectively.

$$P_t = t + (1-t)\beta m/M \tag{2}$$

$$P_f = \beta(1-t)(m/M) \tag{3}$$

Then, it can be approximated that the distribution of S_i which is a neighborhood support measurement of match x_i has a couple of binomial distribution:

$$S_i \sim \begin{cases} B(n, p_t) & if\ x_i\ is\ true \\ B\left(n, p_f\right) & if\ x_i is\ false \end{cases} \tag{4}$$

Thus, it can be separated true and false matches by the score of S and an appropriate *threshold*. Due to the large area of motion smoothing, a more generalized score can be given in (5):

$$S_i = \sum_{i=1}^{K} |\chi_{a^i b^i}| - 1 \tag{5}$$

Where K refers to the number of adjacent disjoint grids in the grid region. And the distributions of S_i are as follows:

$$S_i \sim \begin{cases} B(Kn, p_t) & if\ x_i\ is\ true \\ B\left(Kn, p_f\right) & if\ x_i\ is\ false \end{cases} \tag{6}$$

The mean value and variance of S_i are as follows:

$$\left\{ m_t = Knp_t,\ \ s_t = \sqrt{Knt(1-p_t)} \right\} \qquad if\ x_i\ is\ true \tag{7}$$

$$\left\{ m_f = Knp_f, \qquad s_f = \sqrt{Knt\left(1-p_f\right)} \right\} \qquad if\ x_i\ is\ false \tag{8}$$

Grid framework is adopted to make score computing independent of feature numbers. According to prior knowledge, the image is divided into $G = 20 \times 20$ overlapping cells. In order to improve the robustness, grouping cell-pairs based on a smooth lateral motion

assumption shown in Fig. 3 are used. Making a grid selection by rotation the potential on all potential scales and choosing the best result can solve the problem of image rotation and scale changing, the desired *threshold* can be given as:

$$\tau = m_f + \alpha s_f \tag{9}$$

Where α is an adjusting parameter. In practice, the number of m_f is small, while the value of α is large. Thus, the number of τ can be approximated to:

$$\tau \approx \alpha s_f \approx \alpha \sqrt{n} \tag{10}$$

The algorithm to match feature points is shown in Algorithm 1. Significantly, the input of Algorithm 1 is a pair of images I_a and I_b, while the output is the set of properly matched feature points, *inliers*. Additionally, G denotes the number of grids, $|\chi_{ik}|$ and $|\chi_{ij}|$ represent the matched quantities of the grid, S_i and τ can calculated by Eqs. (6) and (10).

Algorithm 1 feature matching by GMS
Input: One pair of images I_a, I_b
Initialization:
1: Detect feature points and calculate their descriptors
2: For each feature in I_b, find its nearest neighbor in I_a
3: Divide two images by G grids respectively
4: **for** i=1 to G **do**
5: j=1;
6: **for** k=1 to G **do**
7: **if** $\lvert\chi_{ik}\rvert > \lvert\chi_{ij}\rvert$ **then**
8: $j = k$;
9: **end if**
10: **end for**
11: Computer S_i, τ ;
12: **if** $S_i > \tau$ **then**
13: $Inliers = Inliers \cup \chi_{ij}$;
14: **end if**
15: **end for**
Iteration: Repeat from line 4, with grid patterns shifted by half cell-width in the x, y and both x and y directions.
Output: *Inliers*

3.3 Dynamic area detection algorithm

In this sub-section, the principle and pipeline of dynamic area detection method is introduced. Recently, people gradually combine semantic recognition with SLAM to make robots understand surrounding. Many researchers use the neural network to detect every object in the scene, which greatly limits the range of application. Those papers roughly define people or cars as dynamic depends on the test environment, which is partly speculative. It can be assumed that the test environment is in a crowded street, although using the convolution neural network can detect every pedestrian accurately, all these objects are regarded as the dynamic objects and

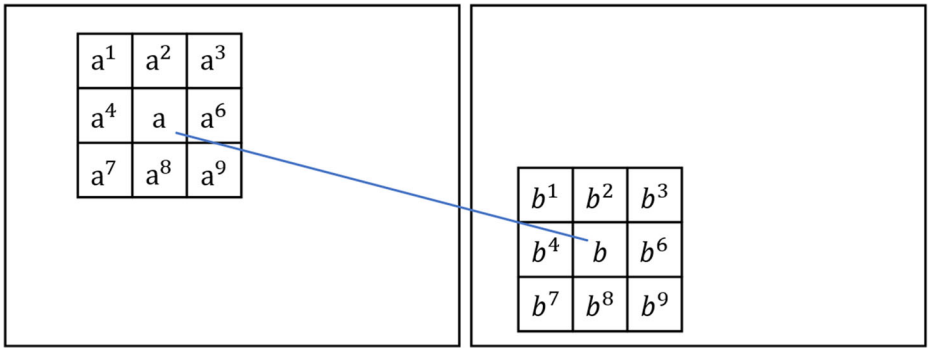

Fig. 3 The basic motion kernel

eliminated out of images. Additionally, these methods often have an assumption that there are enough feature points in the image, and the distribution of feature points is even, which are actually untenable. The dynamic area detection method proposed by this paper ignores these assumptions, breaks through the limitation of target types and realizes dynamic region segmentation without classification.

GMSK method combines GMS feature matching with the k-means cluster method to detect dynamic areas for the first time. GMS feature matching method can improve matching stability and reduce the probability of false matching, while k-means cluster method can automatically divide a bunch of unlabelled data into categories, ensuring that the same kind of data have similar characteristics. Inspired by these advantages, the GMSK dynamic region detection algorithm is proposed.

Given a set of d-dimensional real vector data $\{x_1, x_2, \cdots, x_n\}$. K-means clustering perform partitioning tasks of the n observation into $K(<n)$ set $S = \{S_1, S_2, \cdots S_k\}$ to minimize the within-cluster sum of distance functions of each point in the cluster to K-center. The formula below shows the K-means function:

$$argmin\sum_{i=1}^{k}\sum_{x\in S_i}\|x-\mu_i\|^2 \tag{11}$$

In the feature-based SLAM system, two consecutive frames of the images are generally used for feature matching to perform system initialization and pose tracking. However, in order to better represent the algorithm, in this section, we select two discontinuous images for algorithm simulation. As can be seen from the first row of the Fig. 4, dense feature points were extracted from the image. It can be seen from the figures that the distribution of feature points is dense and uneven. Then the GMS algorithm is used to match the feature points of the image pair. The second row of the Fig. 4 show the matching results of feature points. In the figures, the feature points that match successfully are displayed in red, and the feature points that failed to match are displayed in green. Feature points on moving objects will cause inaccurate estimation of camera pose in the back-end system, so these points need to be pre-processed by algorithms to reduce the impact on the system. It can be found from the second row of the Fig. 4 that on the back of the man in the plaid shirt, there are significantly more green points than red points, the same thing happens in the man's face on the left. Through the above analysis, it can be considered that in these regions, the probability of failure of feature point matching is greater than the probability of success.

Fig. 4 Steps of dynamic area detection algorithm

After detecting and matching all feature points in the images, then we use the sliding windows to statistics the number of unmatched feature points. Firstly, according to prior knowledge, we use 400 grids to divide the images evenly, which aims to divide the whole image into multiple

grids to easily distinguish the dynamic area and static area. We use the sliding window model to realize the statistics of the number of unmatched feature points in every grid. According to the formula (5), S_l is used to represent the score of an unmatched feature point pair in the grid, and $\overline{X_i}$ represents the number of unmatched feature point pairs in the grid.

$$S_l = |X_i| - 1 \tag{12}$$

In every grid, δ can be calculated by using the sliding window model:

$$\delta_i = \frac{N_{un}}{N_m} \times 100\% \tag{13}$$

Here, N_{un} denotes the number of unmatched feature points in the grid, N_m denotes the number of matched feature points in the grid. Through the distribution of feature points in the images, we can roughly distinguish the dynamic region and the static region of the images.

The next step, adopt the k-means cluster algorithm to cluster the number of unmatched feature points in the image. By using the grid to segment the images, it can be known that δ in each grid is representative. In this paper we use the K-means cluster algorithm to divide the dynamic grids and static grids. In the K-means algorithm, assuming that input samples $S = X_1$, X_2, …, X_m, then chose K category center, u_1, u_2, …, u_k. For every sample X_i, define $label_i$ as:

$$label_i = arg \min_{1 \le j \le k} \left\| X_i - u_j \right\| \tag{14}$$

Then, updating each category center to the mean of all samples belonging to that category. At last, repeat until the change in the category center is less than a certain *threshold*. In this paper, since the feature points need to be divided into dynamic regional feature points and static regional feature points, we set K as 2.

K-means algorithm divides the number of unmatched feature points into two parts, one part represents the static grids of the image, another part represents the dynamic region of the image. The third row of the Fig. 4 show the detection results, it can be seen that parts of the two experiments in the image are divided into a single category, and distinguished from the static environment. As shown in the figures, the area in the red box is defined as the dynamic region, which focuses on the people in the images. It can be seen from the raw input figures, there are two experimenters in the pictures. The sitting experimenter has no major posture changes in the two images, while another posture changes greatly. In response to this difference, the proposed algorithm cleverly uses the relationship between the number of feature matches to detect part of the experimenter's body structure with obvious posture changes in the picture as a dynamic area, while retaining other parts. This region-based segmentation algorithm has not been considered and implemented by other algorithms.

The algorithm to detect dynamic area is shown in Algorithm 2. Significantly, the output of Algorithm 2 is a pair of images I_a and I_b, while the output is the dynamic areas groups. Additionally, G of the Step4 denotes the number of grids, N_{um} and N_m represents the quantities of unmatched feature points and matched feature points of

each grid, respectively. The value of δ_i can be calculated by Eq. (13), K denotes that the data will be clustered into two categories, μ_k represents the centre of mass, which starts with a random number.

Algorithm 2 dynamic areas detection by GMSK
Input: One pair of images I_a, I_b
Initialization:
1: Detect feature points and calculate their descriptors
2: Match feature points by GMS algorithm
3: Divide two images by G grids respectively
4: **for** i=1 to G **do**
5: Statics unmatched feature points N_{un};
6: Statics matched feature points N_m;
7: Calculate δ_i;
8: **end for**
9: Set K as 2;
10: Randomly select K data from N_{un} as the center of mass μ_k
11: **for** $i = 1$ to N_{un} **do**
12: **for** $k = 1$ to K **do**
13: distance $(k) = \sqrt{(N_{un}[i].x - \mu_k)^2 + (N_{un}[i].y - \mu_k)^2}$
14: **end for**
15: **if** (distance (1) < distance (2))
16: Divide i to the first group
17: **else**
18: Divide i to the second group
19: **end if**
20: **end for**
Output: the first group, the second group

3.4 Dynamic area rejection

In this sub-section, we combine dynamic area detection results and moving consistency check results to complete the establishment of dynamic area: the area is moving or not moving, and at the same time, remove the feature points of the dynamic area. If there are a certain number of dynamic points produced by moving consistency check fall in the contours of a segmented object, then this area is determined to be moving.

After confirming the dynamic area, the next important problem to be solved is the elimination of feature points among the dynamic areas. The proposed strategy is that if the area in the image is determined to be moving, then remove all the feature points located in the moving area. In this way, outliers can be eliminated precisely. Figure 5 demonstrate the results of dynamic region rejection, the first row of the figures show the distribution of the feature points extracted by the ORB algorithm, the second row show the images after the outliers have been removed. In the images, people in black T-shirt and people in checked shirt belong to dynamic objects. Most of the papers directly remove this part of the feature points. But in the green box area of the figures, the person's legs did not move and should not regard as moving

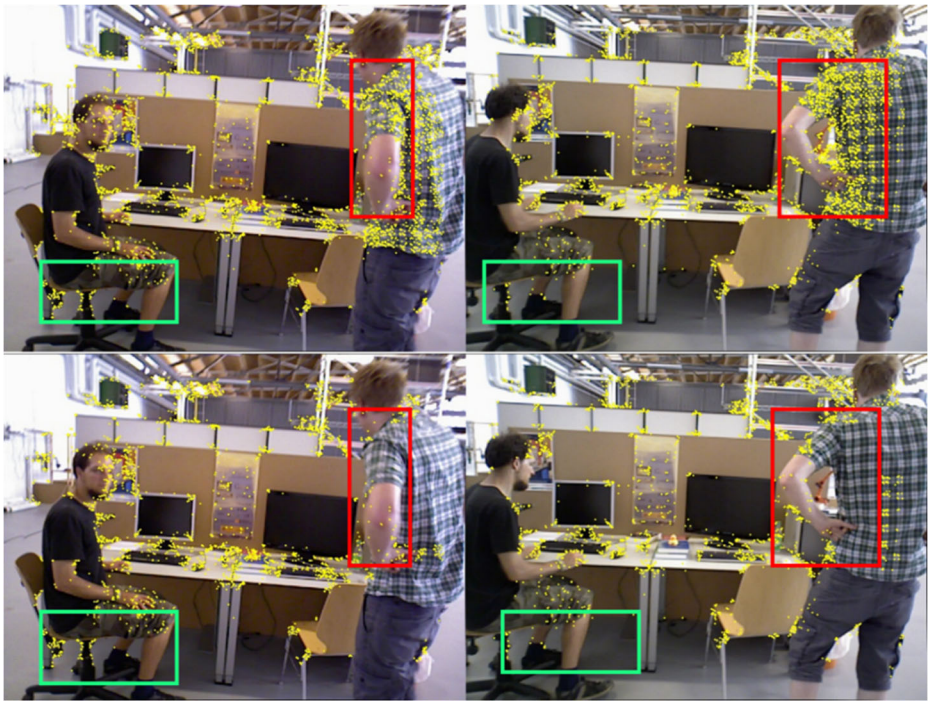

Fig. 5 Comparison of dynamic regional feature points before and after removal

targets. Therefore, the maintenance of feature points in these parts will greatly increase the number of available feature points. The areas selected by the red box represent the moving parts of the dynamic targets. Removing these feature points can effectively reduce the impact of dynamic objects in the scene for camera pose estimation.

4 Results and discussion

In this section, experimental results and related discussion would be presented to demonstrate the effectiveness of the proposed method. Firstly, the symbols used for the experiment are as follows:

- 'GMSK-SLAM' denotes the proposed method in this paper.
- 'ORB-SLAM2' denotes the traditional Simultaneous localization and mapping method, which is described in [25].
- 'The latest method' denotes the state-of-art method proposed by Wang. in 2019, which is described in [38].
- 'Truth' denotes the ground truth obtained by an external motion capture system.
- 'Difference' denotes the difference between the estimated trajectory and the truth trajectory.

Different dynamic environments' sequences in the TUM data set are used to assess the localization accuracy of the presented RGB-D SLAM scheme. Additionally, we evaluate our

presented framework by comparing it with other state-of-the-art SLAM system for better illustrating the effectiveness of the proposed system. The TUM dataset is a scenic dataset of the Technical University of Munich, including 50 laboratory and outdoor sequences. The RGB and depth images were recorded at a frame rate of 30 Hz and a 640 × 480 resolution. Ground-truth trajectories obtained from a high-accuracy motion-capture system are provided in the TUM datasets. The proposed algorithm is operated on a laptop with an Intel i7CPU, 16G of memory, and Ubuntu16.04. GPU acceleration was not adopted during the experiments. Firstly, GMSK-SLAM was compared with the traditional method, ORB-SLAM2, on the TUM fr3_walking_xyz sequence to verify the effectiveness and necessity of detecting dynamic areas and removing outliers. Secondly, to demonstrate the characteristics of high precision for different dynamic environments, the experiments are carried out on more TUM datasets. Finally, compared GMSK-SLAM with a new method proposed by Wang. in 2019 to check if the proposed method is more reliable in dynamic environments.

4.1 Evaluation using TUM RGB-D dataset

The TUM RGB-D dataset [35] provides several sequences in dynamic environments with accurate ground truth obtained with an external motion capture system, such as walking, sitting, and desk. The TUM dataset is divided into high-dynamic datasets and low-dynamic datasets. The desk sequence describes a scene in which a person sits at a desk in an office. In the sitting sequences, two persons sit at a desk with a little gesture. These two scenarios can be considered lowly dynamic environments. In the walking sequences, two persons walk through an office scene. Walking sequences can be used to evaluate the robustness of the proposed method because it is in highly dynamic scenes with quickly moving objects. Since some people walk back and forth in the walking sequences, the walking sequences are mainly used for our experiments. People in these sequences could be regarded as high-dynamic objects, and they are the most difficult problem to deal with. The sitting sequences are also used, but they are considered as low-dynamic sequences as the person in them just moves a little bit.

ORB-SLAM2 is recognized as one of the most outstanding and stable SLAM algorithms at present, so a comparison made between ORB-SLAM2 and GMSK-SLAM is of great significance. Figure 6 shows the trajectory comparison between ORB-SLAM2 and GMSK-SLAM in high dynamic fr3_walking_xyz sequence. In Fig. 6, the blue lines represent the trajectory estimated by ORB-SLAM2 and GMSK-SLAM algorithm, and the black represents the truth trajectory, while lines in red show the difference between estimation and ground truth. The longer the red lines, the

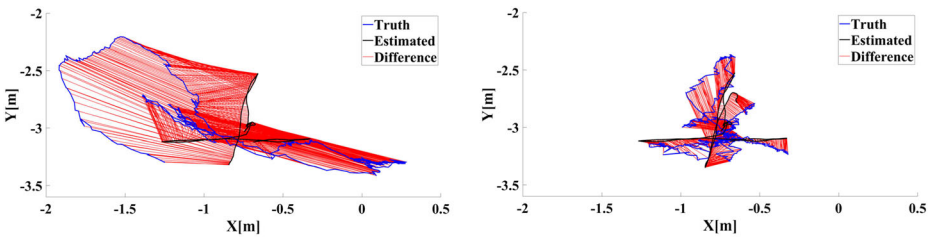

Fig.6 **a** Global trajectory estimated by ORB-SLAM2, **b** Global trajectory estimated by GMSK-SLAM

greater the trajectory error. As can be seen, in the same coordinate scale, due to adding dynamic areas detection thread, the errors are significantly reduced in GMSK-SLAM, the global estimated trajectory shows a tendency of convergence, and the estimated trajectory is much closer to the true trajectory.

After the global trajectory comparison, we further compare the trajectory accuracy of each axis. The ATE (Absolute Trajectory Error), which represents the difference between the estimated trajectory and the ground-truth, is used as the evaluation metric. It directly indicates the localization accuracy of the system. In particular, an easy-to-use open-source package evo is employed for the evaluation (github.com/MichaelGrupp/evo). The Fig. 7 show the estimated trajectories of GMSK-SLAM and ORB-SLAM2 on the highly dynamic datasets. The blue lines and the green lines represent the trajectory of GMSK-SLAM and ORB-SLAM2, respectively. The grey dotted lines represent the ground-truth of the trajectory reference. As can be seen from Fig. 7, the coincidence degree of blue lines and grey lines is larger than that of green lines and grey lines, which means the estimation accuracy of the proposed method plays superior performance than the ORB-SLAM2 in every axe. We also use the evo tool to plot the APE (Absolute Pose Error) of the fr3_walking_xyz sequence in the TUM dataset (Fig. 8). Figure 8 shows the APE distribution of the GMSK-SLAM and ORB-SLAM2 system throughout the tracking. As we can see in the green line of Fig. 8, when there are pedestrians in the field of view of the camera, the APE values increase dramatically. However, the APE values of the blue line at those same moments are greatly reduced, because the influence of pedestrians has been eliminated with our algorithm. In addition to ATE and APE. We also use RMSE to value the positioning accuracy of the whole GMSK-SLAM system.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(P_i - T_i)^2} \tag{15}$$

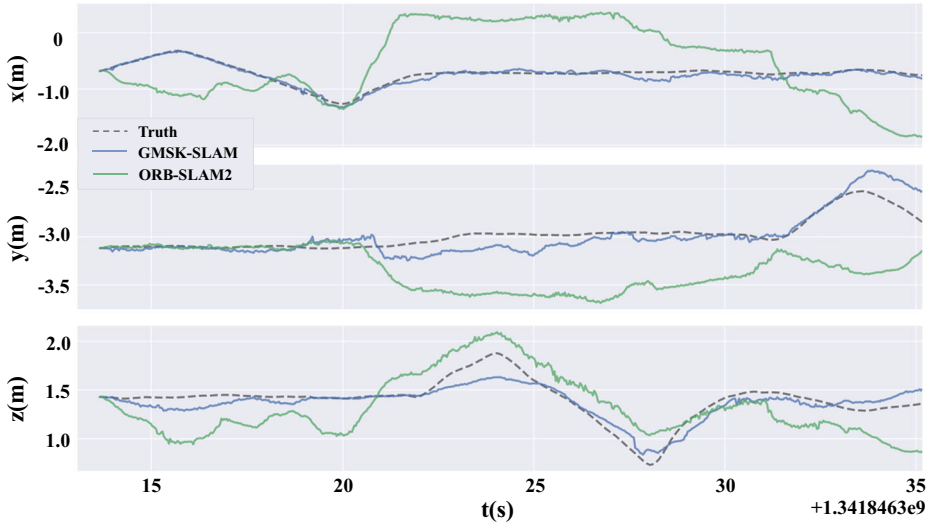

Fig. 7 The trajectory of ORB-SLAM2 and GMSK-SLAM

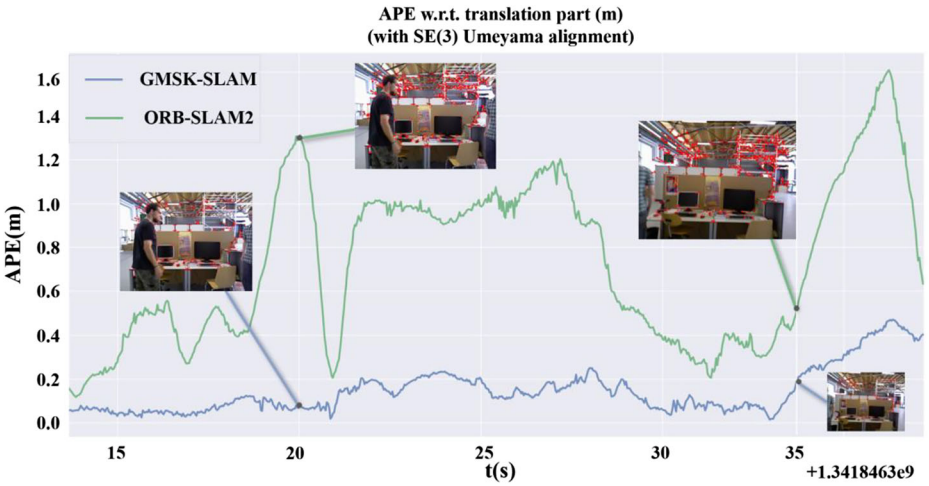

Fig. 8 The Absolute Pose Error results of GMSK-SLAM and ORB-SLAM2

RMSE can evaluate the degree of change in the data, the smaller the RMSE value is, the more accurate the prediction model can be in describing the experimental data. We compute the RMSE of ATE to test the accuracy of the algorithm. In the presented GMSK-SLAM system, the RMSE of ATE of X-axis is only 1.8365×10^{-4}m, Y-axis is 0.0017 m and the Z-axis is 1.9623×10^{-4}m. While, the RMSE of ATE X-axis is 0.2686 m, Y-axis is 0.0156 m, and the Z-axis is 0.04 m with ORB-SLAM2. The above experiments show that the applicability of the GMSK-SLAM algorithm in dynamic environment is greater than that of ORB-SLAM2. To further test the applicability of GMSK-SLAM in different environment, a variety of dynamic environments are selected for testing. The quantitative comparison results are shown in Tables 1 and 2, where xyz, static, rpy, and half in the first column stand for four types of camera ego-motions, for example, xyz represents the camera moves along the x-y-z axes. We present the values of RMSE, Mean Error, and Standard Deviation (S.D.) in this paper, while RMSE and S.D. are more concerned because they can better indicate the robustness and stability of the system. The S.D. values in the tables are calculated as follows:

$$S.D. = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{16}$$

Where μ denotes the mean of x_1, x_2, $\ldots x_N$. The Mean Error in the tables is calculated as follows:

$$Mean = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu) \tag{17}$$

The values of improvement of GMSK-SLAM compared to the original ORB-SLAM2 are also calculated. The improvement is calculated by the following formula:

$$\eta = \frac{\alpha_{ORB} - \alpha_{DMSK}}{\alpha_{ORB}} \times 100\% \tag{18}$$

Where η denotes the value of improvement, α_{ORB} denotes the value of ORB-SLAM2 and α_{DMSK} denotes the value of GMSK-SLAM.

Table 1 RESULTS OF METRIC TRANSLATIONAL DRIFT(RPE)

Sequences	ORB-SLAM2			GMSK-SLAM			Improvements		
	RMSE(m)	Mean(m)	S.D. (m)	RMSE(m)	Mean(m)	S.D. (m)	RMSE	Mean	S.D.
Fr3_walking_xyz	0.4124	0.3110	0.2684	0.0086	0.0049	0.0294	**97.9**%	**98.4**%	**89.0**%
Fr3_walking_static	0.2162	0.0905	0.1962	0.0034	0.0027	0.0018	**98.4**%	**97.0**%	**99.09**%
Fr3_walking_rpy	0.4249	0.2825	0.3166	0.0614	0.0343	0.0075	**85.5**%	**87.9**%	**76.8**%
Fr3_walking_half	0.3550	0.2161	0.2810	0.1155	0.0186	0.0247	**67.29**%	**91.3**%	**91.2**%
Fr3_sitting_static	0.0095	0.0083	0.0046	0.0048	0.0028	0.0017	**49.5**%	**66.3**%	**63.0**%

Table 2 RESULTS OF ABSOLUTE TRAJECTORY ERROR(ATE)

Sequences	ORB-SLAM2			GMSK-SLAM			Improvements		
	RMSE(m)	Mean(m)	S.D. (m)	RMSE(m)	Mean(m)	S.D. (m)	RMSE	Mean	S.D.
Fr3_walking_xyz	0.7521	0.6492	0.3759	0.0360	0.0270	0.0605	**95.2%**	**95.8%**	**83.9%**
Fr3_walking_static	0.3900	0.3554	0.1602	0.0270	0.0280	0.0491	**93.1%**	**92.1%**	**69.3%**
Fr3_walking_rpy	0.8705	0.7425	0.4520	0.0100	0.0160	0.0110	**98.8%**	**97.8%**	**97.5%**
Fr3_walking_half	0.4863	0.4272	0.2290	0.0590	0.0431	0.0698	**87.86%**	**90.0%**	**69.5%**
Fr3_sitting_static	0.0087	0.0076	0.0043	0.0009	0.0008	0.0013	**89.6%**	**89.4%**	**69.7%**

As can be seen from Tables 1 and 2, GMSK-SLAM can make the performance in most high-dynamic sequences get an order of magnitude improvement. In terms of ATE, the RMSE and S.D. improvement values can reach up to 98.8% and 97.5% respectively. The results indicate that GMSK-SLAM can improve the robustness and stability of the SLAM system in high-dynamic environments significantly. Therefore, from tables, it can be found that the accuracy of GMSK-SLAM is significantly greater than ORB-SLAM2, which demonstrates that it is necessary to detect and remove moving objects in dynamic environments. The estimation of camera pose mainly depends on the extraction and matching of feature points for ORB-SLAM2, so optimizing the feature points in dynamic environments would greatly increase the accuracy of attitude and position.

For practical applications, real-time performance is a crucial indicator to evaluate the SLAM system. We test the time required for some major modules to process. The results are shown in Table 3. The average time in the main thread to process each frame is 57.3ms, including dynamic area detection, visual odometry estimation, pose graph optimization. Compared with previous non-real-time methods to filter out dynamic objects, such as [36], GMSK-SLAM is more satisfied with the needs of the real time.

It can be seen from the above analysis, in a variety of complex and changing scenarios, the proposed method in this paper still can maintain high positioning accuracy. Compared with traditional ORB-SLAM2, the RMSE accuracy are all improved almost 90% in different environment. Experiments in various scenarios prove that the proposed algorithm has high accuracy and good stability when dealing with different complex dynamic environments. Therefore, the algorithm in this paper is more suitable for the environment perception and location in the actual dynamic scene.

4.2 Comparison with other method

In this section, the proposed method is compared with a state-of-the-art method, which was proposed by Wang [38] in 2019. This method clustered the filled depth images and used them to segment moving objects. The latest method treats people as dynamic in any situation, even

Table 3 TIME EVALUATION

Module	ORB feature extraction	Moving consistency check	Dynamic area detection
Thread	Tracking	Tracking	Dynamic area detection
Time(ms)	9.132587	25.56298	40.57496

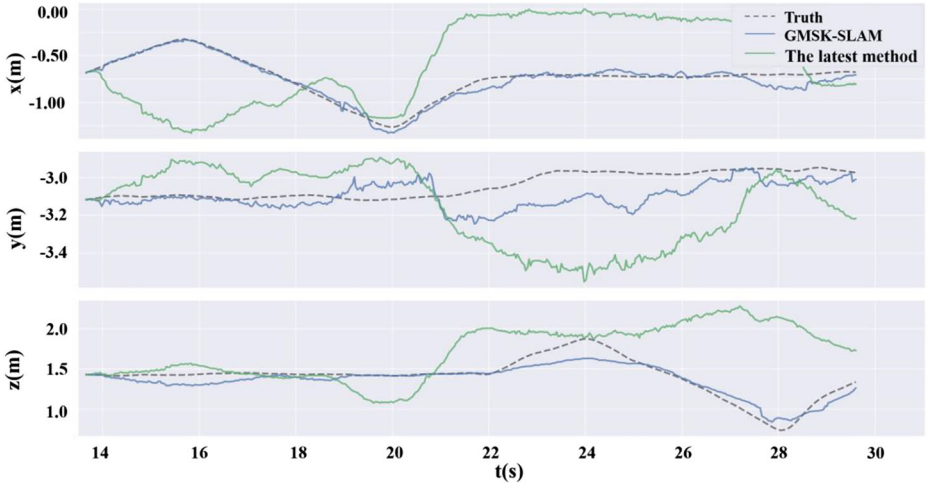

Fig. 9 The trajectory of The Latest method and GMSK-SLAM

the person is in a static state. However, the proposed GMSK-SLAM innovatively detects dynamic areas rather than dynamic objects, determining dynamic or static area depends on the number of matching feature points. Such a method is more realistic. The GMSK-SLAM algorithm maintains the static feature points in dynamic objects effectively, which greatly increases the number of useful feature points. A qualitative comparison between the proposed GMSK-SLAM algorithm and the latest method for dynamic environment is provided in Fig. 9. These quantitative evaluations were carried out using the TUM RGB-D datasets.

It can be seen by comparing Figs. 7 and 9 that removing dynamic objects feature points indeed can effectively improve the accuracy of trajectory. As is showed in Fig. 9, although the latest method reduces the difference between the ground truth and estimated trajectory to some extent, due to removing all feature points in dynamic objects, instead, it also reduces the number of feature points available. Figure 9 clearly demonstrates that the trajectory estimation results of the detecting dynamic region are better than that of the dynamic objects.

The RMSE of ATE in three axes are calculated as follows. As can be seen in Table 4, the proposed GMSK-SLAM has a more reliable performance than the latest method [38]. The comparison shows that the GMSK-SLAM gives greatly better results than the latest method.

In order to better illustrate the effectiveness of our algorithm, we also compare the APE with the latest method. Figure 10 shows the APE distribution of the GMSK-SLAM and the latest method throughout the tracking. As can be seen from the Fig. 10, the proposed method's accuracy in the Fr3_walking_xyz is better than the latest SLAM system.

Table 4 RESULTS OF ABSOLUTE TRAJECTORY ERROR(ATE)

Fr3_walking_xyz	RMSE-X (m)	RMSE-Y (m)	RMSE-Z (m)
The latest method	9.8994×10^{-4}	0.0816	0.0021
The proposed method	$\mathbf{1.8365\times10^{-4}}$	**0.0017**	$\mathbf{1.9623\times10^{-4}}$
Improvements	**81.45%**	**97.92%**	**90.66%**

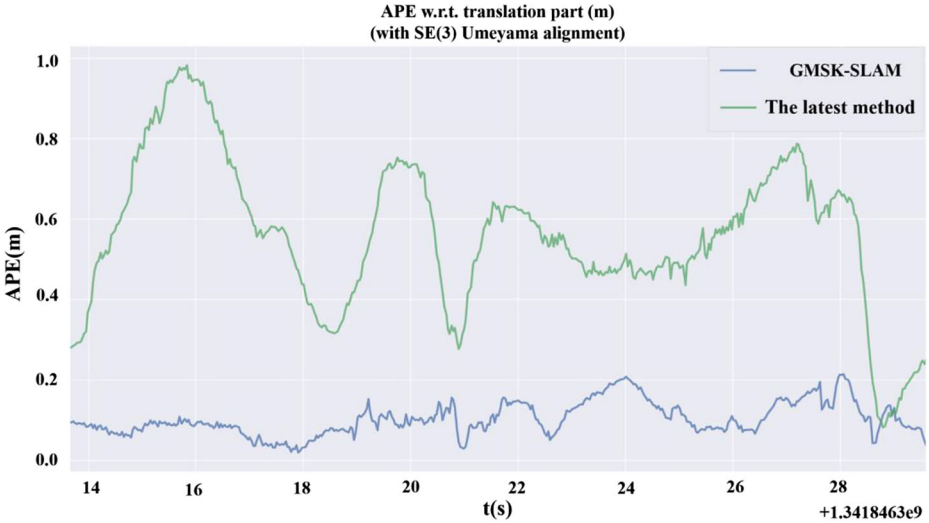

Fig. 10 The Absolute Pose Error results of GMSK-SLAM and the latest method

Finally, a global comparison is made among the three algorithms in Fig. 11. It can be seen from Fig. 11, compared with the ORB-SLAM2, the position error of the GMSK-SLAM and the latest method are reduced to a certain extent. However, the latest method directly eliminates all feature points in dynamic target, which in turn reduces the number of feature points reliable. Therefore, the accuracy of the latest method is lower than the GMSK-SLAM's.

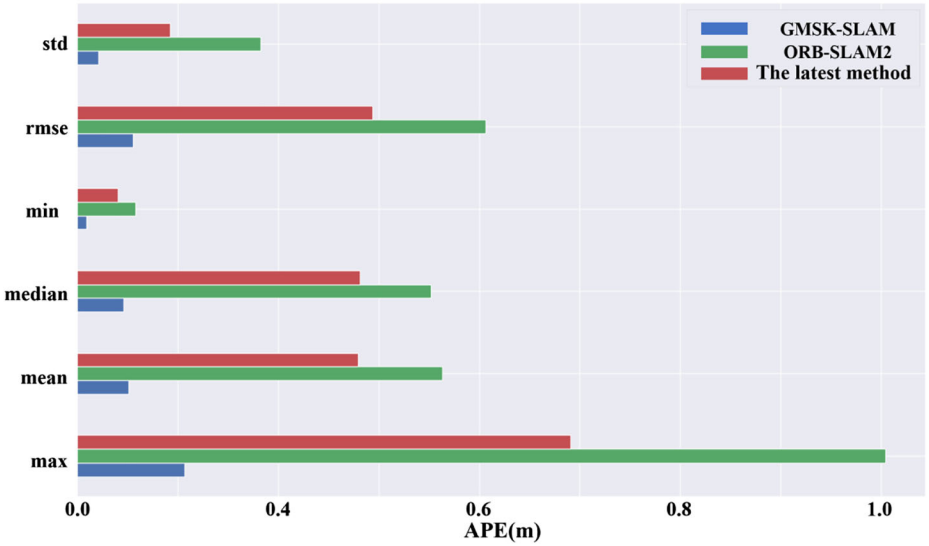

Fig. 11 The comparison of GMSK-SLAM, ORB-SLAM2 and the latest method

5 Conclusion

In this paper, a new complete robust dynamic area detection SLAM (GMSK-SLAM) system is proposed, which could greatly reduce the influence of dynamic objects on pose estimation. The proposed method has third innovations. First, adopting moving consistency check into the proposed method, which greatly increase the accuracy of dynamic objects detection. Second, using the GMS algorithm to match feature points and k-means algorithm to distinguish dynamic environments and static environments. Thirdly, adding a dynamic area detection in an independent thread, which greatly improves the operating efficiency and robustness of the system. The proposed system could run the tracking thread and the dynamic area detection thread in parallel, thus allowing our system to read read-time performance.

In the experimental section, the proposed GMSK-SLAM is qualitatively evaluated with the public benchmark data set TUM. On the one hand, the GMSK-SLAM is compared with the mainstream VIO system, which validly proofs that detecting dynamic area can improve the environmental applicability and stability of the algorithm. All tests show that the trajectory accuracy of the GMSK-SLAM is 90% better than the ORB-SLAM method. On the other hand, a series of experiments are conducted between GMSK-SLAM and the latest method proposed in 2019. Experiments effectively demonstrate that removing dynamic areas rather than dynamic objects significantly eliminates the interference of dynamic factor to the system. Above all, we can conclude that the proposed GMSK-SLAM algorithm has better performance than the traditional method and latest method.

In the future, we plan to consider other environmental disturbance elements, such as environmental brightness, weather conditions, and dynamic target density, etc. Besides, we intend to combine our moving areas detection method with a lightweight deep learning method, to achieve robust results of moving areas detection in challenging dynamic environments.

Acknowledgments This work was supported in part by National Natural Science Foundation of China 52071080, Fundamental Research Funds for the Central Universities under Grant 2242021K1G008, Remaining funds cultivation project of National Natural Science Foundation of Southeast University under Grant 9S20172204.

Code available No (Not applicable).

Data availability Yes

Declarations

Conflicts of interest/Competing interests No

References

1. Badrinarayanan V, Kendall A, Cipolla R (2017) Segnet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell 39(12):2481–2495
2. Bahraini MS, Bozorg M, Rad AB (2018) SLAM in dynamic environments via ML-RANSAC. Mechatronics 49:105–118
3. Bay H (2006) Surf: speeded up robust features. 9th European Conference on Computer Vision (ECCV 2006), Graz, AUSTRIA, pp 404–417

4. Bian JW, Lin WY, Matsushita Y (2017) GMS: Grid-Based Motion Statistics for Fast, Ultra-Robust Feature Correspondence. 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, pp 2828–2837
5. Calonder M, Lepetit (2010) Brief: binary robust independent elementary features. 11th European Conference on Computer Vision, Heraklion, GREECE, pp 778–792
6. Davison AJ, Reid ID, Molton ND (2007) MonoSLAM: real-time single camera SLAM. IEEE Trans Pattern Anal Mach Intell 29(6):1052–1067
7. Dissanayake MWMG, Newman P (2013) A solution to the simultaneous localization and map building (slam) problem. IEEE Trans Robot Autom 17(3):229–241
8. Endres F, Hess J, Engelhard N (2012) An evaluation of the RGB-D SLAM system. IEEE international conference on robotics and automation (ICRA), St Paul, MN, pp 1691-1696
9. Engel J, Schöps T, Cremers D (2014) Lsd-slam: large-scale direct monocular slam. In: proceedings of European conference on computer vision (ECCV), vol 8690, pp 834-849
10. Engel J, Koltun V, Cremers D (2018) Direct sparse Odometry. IEEE Trans Pattern Anal Mach Intell 40(3): 611–625
11. Fang Y, Dai B (2009) An improved moving target detecting and tracking based on optical flow technique and Kalman filter. 4th International Conference on Computer Science and Education, Nanning, PEOPLES R CHINA, pp 1197–1202
12. Forster C, Pizzoli M, Scaramuzza D (2014) SVO: Fast Semi-Direct Monocular Visual Odometry. IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, PEOPLES R CHINA, pp 15–22
13. Harris C G, Stephens M J (1988) A combined corner and edge detector. Proceedings of the 4th Alvey vision conference, Manchester, England, pp 147-151
14. Hess W, Kohler D, Rapp H (2016) Real-time loop closure in 2D LIDAR SLAM. IEEE international conference on robotics and automation (ICRA), pp 1271-1278
15. Klein G, Murray D (2007) Parallel tracking and mapping for small AR workspaces. IEEE & Acm International Symposium on Mixed & Augmented Reality.
16. Kohlbrecher S, Stryk OV, Meyer J (2011) A flexible and scalable SLAM system with full 3D motion estimation. IEEE International Symposium on Safety, Security, and Rescue Robotics, Kyoto, Japan https://doi.org/10.1109/SSRR.2011.6106777
17. Lee SJ, Hwang SS (2019) Bag of sampled words: a sampling-based strategy for fast and accurate visual place recognition in changing environments. Int J Control Autom Syst 17(10):2597–2609
18. Li JN, Wang LH, Li Y (2016) Local optimized and scalable frame-to-model SLAM. Multimed Tools Appl 75(14):8675–8694
19. Liu GH, Zeng WL, Feng B, Xu F (2019) DMS-SLAM: a general visual SLAM system for dynamic scenes with multiple sensors. SENSORS 19(17)
20. Long J, Shelhamer E, Darrell T (2015) Fully Convolutional Networks for Semantic Segmentation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, pp 3431–3440
21. Lowe D (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
22. MacQueen J (1965) Some Methods for Classification and Analysis of Multi-Variate Observations. Proceedings of the Fifth Berkeley Symposium on Math, Statics, and Probability, vol 1, pp 281–297
23. Mu X, He B, Zhang X (2019) Visual navigation features selection algorithm based on instance segmentation in dynamic environment. IEEE Access 8:465–473
24. Muja M, Lowe DG (2009) Fast approximate nearest neighbors with automatic algorithm configuration. VISAPP, vol 1:331–340
25. Mur-Artal R, Tardos JD (2017) ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras. IEEE Trans Robot 33(5):1255–1262
26. Mur-Artal R, Montiel JMM, Tardós JD (2015) ORB-SLAM: A Versatile and Accurate Monocular SLAM System. IEEE Trans Robot 31(5):1147–1163
27. Oh S, Hahn M, Kim J (2015) Dynamic EKF-based SLAM for autonomous mobile convergence platforms. Multimed Tools Appl 74(16):6413–6430
28. Redmon J, Farhadi A (2018) Yolov3: an incremental improvement.arXiv e-prints,2018
29. Redmon J, Divvala S, Girshick R, Farhadi A (2015) You only look once: unified, real-time object detection. 2016 IEEE conference on computer vision and pattern recognition (CVPR), Seattle, WA, pp. 779–788
30. Rosten E, Drummond T (2006) Machine learning for high-speed corner detection. 9th European conference on computer vision (ECCV 2006), Graz, AUSTRIA, pp 430-443
31. Rublee E, Rabaud V, Konolige K et al (2012) ORB: an efficient alternative to SIFT or SURF. IEEE international conference on computer vision (ICCV), Barcelona, SPAIN, pp 2564-2571
32. Saputra MRU, Markham A, Trigoni N (2018) Visual SLAM and structure from motion in dynamic environments: a survey. ACM Comput Surv 51(2):1–36

33. Sharma K (2018) Improved visual SLAM: a novel approach to mapping and localization using visual landmarks in consecutive frames. Multimed Tools Appl 77(7):7955–7976
34. Smith RC, Cheeseman P (1986) On the representation and estimation of spatial uncertainty. Int J Robot Res 5(4):56–68
35. Sturm J, Engelhard N, Endres F (2012) A benchmark for the evaluation of RGB-D SLAM systems. 25th IEEE\RSJ International Conference on Intelligent Robots and Systems (IROS), Algarve, PORTUGAL, pp 573–580
36. Sun Y, Liu M, Meng QH (2017) Improving RGB-D SLAM in dynamic environments: a motion removal approach. Rob Auton Syst 89:110–122
37. Tong Q, Peiliang L, Shaojie S (2018) VINS-mono: a robust and versatile monocular visual-inertial state estimator. IEEE Trans Robot 34(4):1004–1020
38. Wang R, Wan W, Wang Y (2019) A new RGB-D SLAM method with moving object detection for dynamic indoor scenes. Remote Sens 11(10)
39. Wrobel B P (2001) Multiple view geometry in computer vision. Cambrige university press
40. Yu C, Liu Z, Liu X (2018) DS-SLAM: a semantic visual SLAM towards dynamic environments. 25th IEEE/RSJ international conference on intelligent robots and systems (IROS), Madrid, SPAIN, pp 1168-1174
41. Zhang W, Chen Q, Zhang W, He X (2018) Long-range terrain perception using convolutional neural networks. Neurocomputing 275:781–787