



Visual odometry algorithm based on geometric prior for dynamic environments

Gang Xu^{1,2} · Ze Yu² · Guangxin Xing² · Xingyu Zhang² · Feng Pan¹

Received: 17 January 2022 / Accepted: 10 April 2022 / Published online: 25 May 2022
© The Author(s) 2022

Abstract

Simultaneous localization and mapping (SLAM) is considered to be an important way for some smart devices to perform automatic path planning, and many successful SLAM systems have been developed in the past few years. Most existing approaches rely heavily on static world assumptions, and such strong assumptions limit the application of most vSLAM (visual SLAM) in complex dynamic reality environments, where dynamic objects often lead to incorrect data association in tracking, which reduces the overall accuracy and robustness of the system and causes tracking crashes. The dynamic objects in the map may change over time; thus, distinguishing dynamic information in a scene is challenging. In order to solve the interference problem of dynamic objects, most point-based visual odometry algorithms have concentrated on feature matching or direct pixel intensity matching, disregarding an ordinary but crucial image entity: geometric information. In this article, we put forward a novel visual odometry algorithm based on dynamic point detection methods called geometric prior and constraints. It removes the moving objects by combining the spatial geometric information of the image and depends on the remaining features to estimate the position of the camera. To the best of our knowledge, our proposed algorithm achieves superior performance over existing methods on a variety of public datasets.

Keywords Simultaneous localization and mapping · Visual odometry · Dynamic objects

1 Introduction

Since the twenty-first century, simultaneous localization and mapping (SLAM) have attracted great interest because of its potential applications in robot navigation, 3D reconstruction, and autonomous vehicles [1–3]. In SLAM, multiple visual sensors were used to obtain the relevant image sequences. The posture of the objectives (robot, human, or automobile) was estimated by analyzing the images. Some sensors, for example, RGB-D cameras [4], binocular cameras [5], and LiDAR [6], can provide depth information for each image frame, facilitating state estimation, and mapping. Currently, most visual odometers are implemented in a static environment [7]. In the presence of many dynamic objects in the scene, the SLAM fails to perform well, limiting its applications in

actual scenarios. Therefore, in this paper, we aimed to make visual odometers more accurate in dynamic scenarios.

Traditionally, there are two general methods of visual range measurement: feature-based visual odometry (FVO) and dense visual odometry (DVO). The FVO, such as PTAM [8], RGB-D SLAM [9], and ORB-SLAM [10], generate sparse 3D maps for posture estimation based on feature point extraction and matching by minimizing geometric reprojection errors. Recently, the DVO [10, 11] has become prevalent. This method acts directly on the original pixel intensity by minimizing photometric errors. According to Akinlar and Topal [12], a dense or semi-dense map can be generated with more image information, and the heavy geometric projection error of the key points is usually robust to the image noise and the larger geometric distortion and motion. However, the existing SLAM algorithms suffer from poor robustness, the low-texture environment as there are only a few significant features. Generated sparse or semi-dense maps convey little information about motion planning. Although some studies use a plane or scene to regularize the map, they need to get good state estimation from other sources. Li et al. [13] presents a semantic-assisted visual inertial odometry (VIO) system for low-texture scenes and highly dynamic environments.

✉ Feng Pan
pan_feng_63@163.com

¹ Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Jiangnan University, Wuxi, China

² Key Laboratory of Advanced Perception and Intelligence Control of High-End Equipment (Ministry of Education), Anhui Polytechnic University, Wuhu, China

The trained U-shaped mesh will be used to detect moving objects, and its performance in dynamic environments is improved by removing feature points on dynamic objects. The joint optimization of IMU measurement error and reprojection error ensures that the system obtains good pose calculation results in low-texture environments, but the semantic segmentation process leads to system speed reduction. In Engel et al. [14], a direct and sparse model was proposed in the form of a single-order vision-theodolite algorithm, but the 3D model was denser, and the complexity was increased. Ban et al. [15] demonstrated a learning visual odometry (L-VO) and dense 3D mapping, where the system trains deep neural networks in a supervised or self-supervised manner to achieve end-to-end estimation of pose states.

In Costante and Ciarfuglia [16], a new monocular camera ego-motion estimation network architecture LS-VO is proposed. This architecture consists of two branches that jointly learn the potential spatial representation of field inputs and camera motion estimation. The method was tested on datasets KITTI and Malaga, optimizing the robustness of domain transform appearance and dynamic range, but the performance degradation due to excessive fitting limited the entire network.

Despite the advantages of these methods, the dynamic object can still cause a large depth error in the actual environment, preventing the existing methods to estimate camera pose effectively. Researchers have conducted studies towards detecting, recognizing, and eliminating moving objects to solve this problem. For example, Sun et al. [17] detected the edges of moving objects by the variation of pixel intensity between two frames. In this approach that they proposed, the dynamic object points were divided by the clustering of the depth map. The performance of this method was stable in dynamic scenarios, but the real-time performance was rather poor. Wei et al. [18] proposed GMSK-SLAM, which innovatively combines a grid-based motion statistics (GMS) feature point matching method with a K-means clustering algorithm to distinguish dynamic regions from images and retain static information from dynamic environments; it can effectively increase the number of reliable feature points and retain more environmental features; the method can achieve a high improvement of localization accuracy in dynamic environments. However, it can be disturbed by environmental factors such as ambient brightness, weather conditions, and dynamic target density. Importantly, as the line features are more abundant in the structured environment and less affected by the dynamic object, the algorithms based on line features [19, 20] attracted more attention. Yang and Scherer [21] implemented direct monocular odometry using points and lines. They used line features to eliminate dynamic targets in the scene, thus improving the accuracy of visual odometry in the dynamic scene. Kim and Kim [22] built the static background environment by utilizing the depth disparity of previous frames. In a dynamic environment, the

approach enhances the stability of visual odometry. However, when the moving object is parallel to the camera plane, as it was the border of the moving object that is recognized, the impact of the moving object cannot be totally erased. Cheng et al. [23] have leveraged the recent success of deep neural networks for detecting the moving objects, offering a label for each identified object and calculating pre-dynamic weights to account for the possibility of object mobility. Despite its good performance, this method still has the problem of tracking loss. In a low-texture environment, where the dynamic regions take up the majority of the image, the lack of information will cause the tracking process to crash. The reprojection information of feature points is utilized to create an adaptive index for distinguishing dynamic points in Cheng et al. [24], which presented a visual SLAM technique integrating optical flow with semantic masking; it performs well in highly dynamic surroundings, but there is a limitation; if all scenes are dynamic and lack static features, this method cannot obtain accurate results.

In current algorithms research, the three sets of feature points in computing the fundamental matrix may contain mis-matched or dynamic feature points by using the P3P algorithm [25] to estimate the camera and cause the P3P algorithm [25] to fail. In this work, we presented a new framework of RGB-D visual odometry using image geometric information dynamic targets that were eliminated by calculating the similarity between two sets of image matching points. It improved the P3P algorithm [25] and made it suitable for dynamic scenarios. Our method significantly shrinks the errors in the frame tracking and enhances the precision and robustness of the visual odometer when compared to current approaches based on ORB [26].

The rest of this paper is structured as follows. Section 2 briefly describes the related work on visual odometry. Section 3 gives the proposed methodology and makes a specific analysis. The experimental results are shown and analyzed in Sect. 4. Finally, we present a brief discussion and conclusion of this paper in Sect. 5.

2 Methodology

Our algorithm is an RGB-D SLAM based on ORB feature points. In this section, we first introduced feature matching based on triangular geometric constraints and then tracked the keyframes using the P3P algorithm [25] to improve RGB-D SLAM's [4] tracking and mapping ability in dynamic scenarios.

2.1 Feature matching algorithm

In our study, we use ORB [26] feature points to extract features from the image and then match the two contiguous keyframes.



Fig. 1 ORB feature point extraction on TUM dynamic dataset

In the image matching of dynamic scenarios, there may be some feature points of dynamic objects that could greatly affect the estimation of camera pose. The dynamic target matching is shown in Fig. 1. There are moving objects (people) in the figure. To prevent these dynamic points from affecting the accuracy of camera estimation, we designed a way to exclude these dynamic points by using the spatial information of the image.

No matter how the camera moves, the triangle formed by any three fixed points in space is fixed, so the triangles formed by these three points in different camera coordinate systems are similar, as shown in Fig. 2, where the cube represents the camera coordinate systems and the triangle represents the imaging in the camera. Through RGB-D images, we can get partial 3D feature points and 2D feature points (Kinect camera may lose part of the depth information). Therefore, in this paper, we evaluated the nature of feature points (dynamic or static) by comparing the similarity of the triangle enclosed by three sets of feature points in two keyframes.

2.2 Tracking algorithm

Tracking is used to solve the problem of camera pose estimation. RGB-D SLAM [4] uses multiple sets of 3D matching

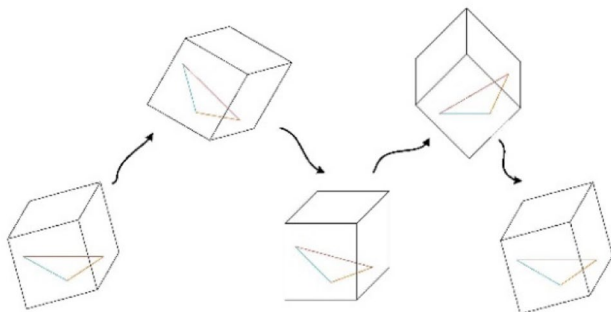


Fig. 2 Triangular structure. The change of the triangle formed by any three points in space in different camera coordinates

points in the two images to estimate the movement of the camera. But it can calculate the ideal pose only if the matching point is completely accurate. The problem is more critical in dynamic scenarios, so we used spatial geometric constraints to restrict these dynamic points. In this work, we used the similarity of triangles to determine that all three feature points are static points to improve the accuracy of the P3P algorithm [25], as shown in Fig. 3.

In Fig. 3, o and \hat{o} represent the origin of the camera’s coordinate system in different poses, and $q_1, q_2,$ and q_3 are three points in space. R and t represent the motion transformation from o coordinate system to \hat{o} coordinate system, where R is the rotation transformation matrix and t is the translation transformation matrix. We know the spatial position of three points in the o coordinate system and the space coordinate system, and we also know their 2D position in the \hat{o} coordinate system. When the position of three spatial points remains unchanged, the P3P algorithm [25] can be used to obtain their accurate 3D coordinates in the \hat{o} coordinate system. At this stage, the triangle formed by the three points in the two-camera coordinate systems is similar. However, when the position of the feature point is changed, the triangle in the \hat{o} coordinate system will be simultaneously changed so that the two sets of triangles are no longer similar.

In the experiment, we evaluated the similarity of two triangles by the ratio of three sides. In the o coordinate system, the three sides of the triangle are respectively:

$$|q_1q_2| = \sqrt{(x_{q_2} - x_{q_1})^2 + (y_{q_2} - y_{q_1})^2 + (z_{q_2} - z_{q_1})^2} \quad (1)$$

$$|q_2q_3| = \sqrt{(x_{q_3} - x_{q_2})^2 + (y_{q_3} - y_{q_2})^2 + (z_{q_3} - z_{q_2})^2} \quad (2)$$

$$|q_3q_1| = \sqrt{(x_{q_1} - x_{q_3})^2 + (y_{q_1} - y_{q_3})^2 + (z_{q_1} - z_{q_3})^2} \quad (3)$$

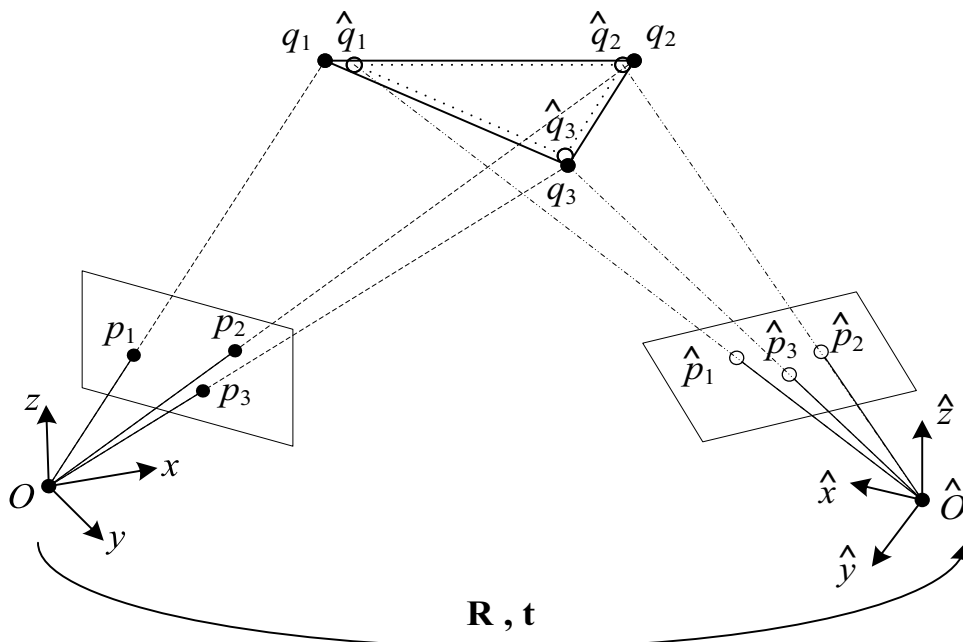
Also, in the \hat{o} coordinate system, the three sides of another triangle are respectively:

$$|\hat{q}_1\hat{q}_2| = \sqrt{(x_{\hat{q}_2} - x_{\hat{q}_1})^2 + (y_{\hat{q}_2} - y_{\hat{q}_1})^2 + (z_{\hat{q}_2} - z_{\hat{q}_1})^2} \quad (4)$$

$$|\hat{q}_2\hat{q}_3| = \sqrt{(x_{\hat{q}_3} - x_{\hat{q}_2})^2 + (y_{\hat{q}_3} - y_{\hat{q}_2})^2 + (z_{\hat{q}_3} - z_{\hat{q}_2})^2} \quad (5)$$

$$|\hat{q}_3\hat{q}_1| = \sqrt{(x_{\hat{q}_1} - x_{\hat{q}_3})^2 + (y_{\hat{q}_1} - y_{\hat{q}_3})^2 + (z_{\hat{q}_1} - z_{\hat{q}_3})^2} \quad (6)$$

Fig. 3 triangle constraint. The state of a moving object in different camera coordinates



Finally, by finding out whether two triangles are similar, we can distinguish whether three feature points are dynamic or static.

$$\frac{q_1q_2}{\hat{q}_1\hat{q}_2} = \frac{q_2q_3}{\hat{q}_2\hat{q}_3} + e_1 = \frac{q_3q_1}{\hat{q}_3\hat{q}_1} + e_2 \tag{7}$$

where e_1 and e_2 represent the conversion errors of q_2q_3 and q_3q_1 from the o coordinate system to the \hat{o} coordinate system respectively, and the total error is:

$$e = e_1 + e_2 \tag{8}$$

When e is less than the set similarity threshold, it is considered that the two groups of triangles are similar, and all the feature points are static; otherwise, the feature points contain dynamic points and need to be selected again.

In this way, we can effectively ensure that the feature points used in every calculation of camera pose are fixed points, thus improving the accuracy of RGB-D SLAM [4] in dynamic scenarios.

3 Experimental results

We conducted our experiments on TUM public dynamic dataset [27]. In the Sitting_xyz dataset and Walking_xyz dataset, the camera keeps facing the desk. The mutual movement of the camera and the person was different from the mimic typical datasets of dynamic scenarios.

3.1 The selection of the similarity threshold

For dynamic scene sequences, we used the ORB algorithm to describe and match feature points, as shown in Fig. 4.

Figure 4a, b show the matching of feature points in the moving scene, in which the triangles formed by two fixed points (3 points) are similar; on the contrary, the triangles formed by two groups of moving points (3 points) are not similar. To facilitate the calculation, we respectively selected 5 sets of matching feature points with depth information to generate 10 different triangles in the two scenarios, four matching points for the fixed point, and the remaining for the dynamic matching points. So, we had four triangles surrounded by fixed points, and one vertex of the remaining six triangles was a dynamic point. The experimental results are shown in Tables 1 and 2.

Table 1 shows that the similarity error of two groups of triangles was less than 0.5 m without dynamic vertices. However, in Table 2, the two sets of triangle errors with dynamic vertices were mostly greater than 0.5. Moreover, the similarity error of the low dynamic scene was smaller than that of the high dynamic scene because of the small change of moving objects in the low dynamic scenario. Finally, we used 0.5 as the similarity threshold to distinguish whether the two triangles were similar.

3.2 Comparison to the prior feature extraction methods

In the selection of feature points, the distance between any two points should be greater than a certain range to avoid three points on the same object.



Fig. 4 Feature point matching. ORB characteristics match on TUM dynamic dataset

Table 1 The similarity error of the triangle without dynamic points

Similarity error (m)	Error 1	Error 2	Error 3	Error 4
Walking_xyz	0.246	0.339	0.273	0.286
Sitting_xyz	0.327	0.259	0.192	0.247

To demonstrate the performance of the proposed algorithm in dynamic scenarios, we used the relative trajectory error as the evaluation index to compare this method with the ORB point feature method and line feature method. The experimental results are displayed in Fig. 5.

Table 2 The similarity error of the triangle with dynamic points

Similarity error (m)	Error 1	Error 2	Error 3	Error 4	Error 5	Error 6
Walking_xyz	0.831	0.749	0.624	0.726	0.663	0.547
Sitting_xyz	0.659	0.497	0.528	0.447	0.594	0.601

Figure 5 shows that the feature point matching algorithm based on spatial triangle constraint had outperformed better than the ORB point feature method in dynamic scenarios.

3.3 TUM dataset evaluation

In order to further prove that the proposed algorithm can effectively improve the robustness and accuracy of SLAM algorithm in dynamic sequences, the experimental results before and after the algorithm improvement are shown in Fig. 6. Figure 6a, b are the comparison diagrams of the real trajectory before the algorithm’s improvement and the

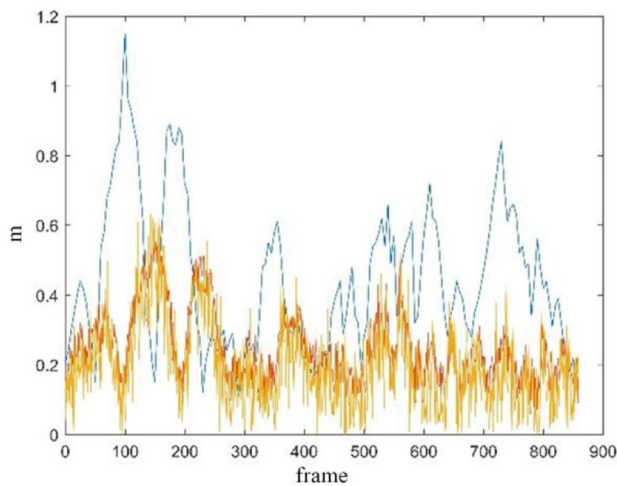
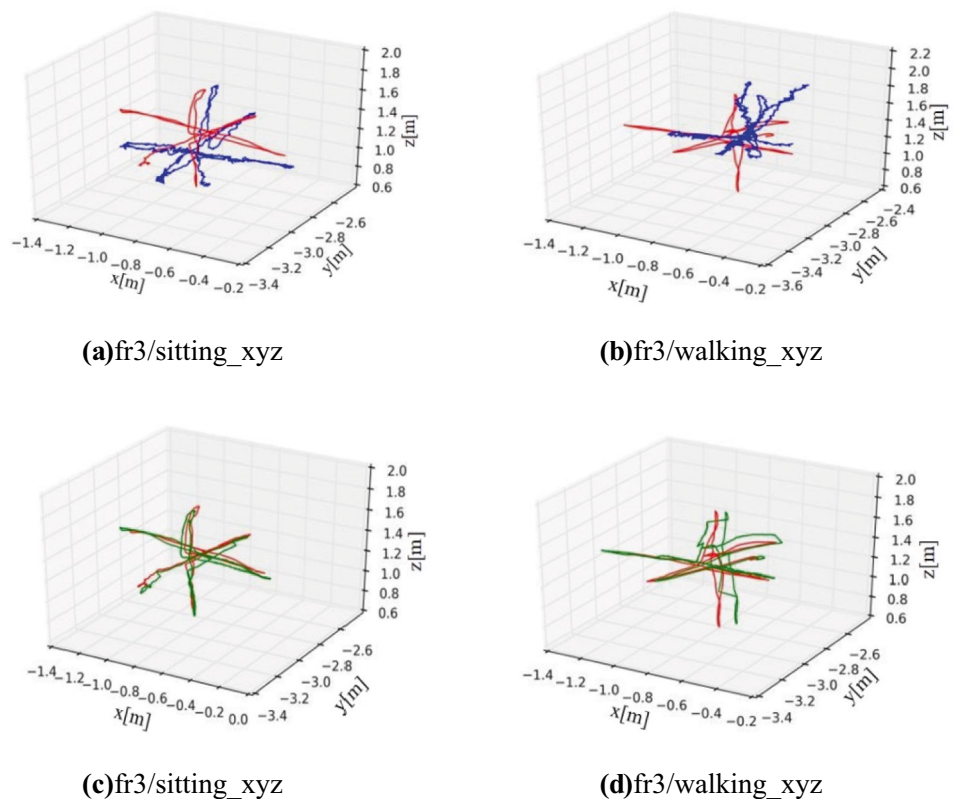


Fig. 5 Error comparison on fr3/Walking_xyz sequence. The blue line represents the visual odometer error using ORB feature matching, The red line represents the visual odometer error using line features and the yellow line represents the visual odometer error of ORB characteristics with triangular structural constraints

experimental trajectory. The red line represents the real pose curve and the blue line represents the pose curve estimated by the algorithm. Figure 6c, d show the comparison between the improved real trajectory of the algorithm and

Fig. 6 Comparison of experimental trajectories and real trajectories of the algorithm before the improvement in different datasets. **a** fr3/Sitting_xyz. **b** fr3/Walking_xyz. **c** fr3/Sitting_xyz. **d** fr3/Walking_xyz



the experimental trajectory. The red line represents the real pose curve and the green line represents the estimated pose curve of the algorithm. Experimental results show that the improved algorithm can effectively reduce the error of camera pose estimation and improve the accuracy of the algorithm in both low dynamic environment and high dynamic environment.

The error in the frame tracking process was significantly reduced, and the accuracy was improved. Figure 7 shows the comparison between the experimental trajectory and the real trajectory tested by the algorithm in this paper on the real scene, demonstrating that the proposed algorithm can better estimate the camera's track in dynamic scenarios.

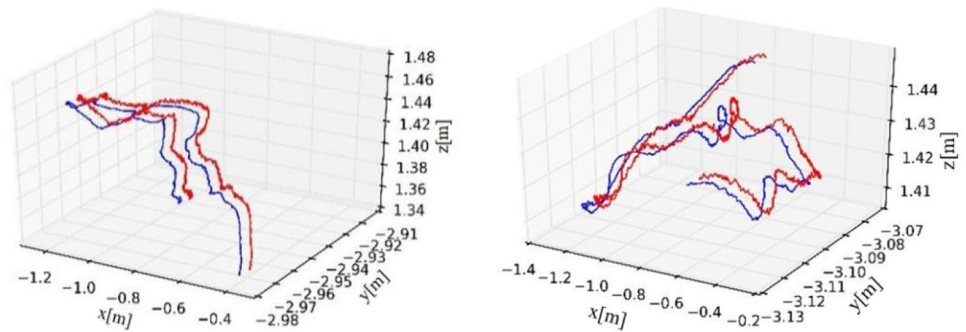
3.4 Evaluation on the complexity

The geometric prior algorithm proposed in this paper ensures that the mis-matching and dynamic feature points in the three sets of feature points are eliminated, and then the P3P algorithm [25] is used when estimating. Compared with the traditional P3P algorithm [25], a total of eight steps from Eqs. (1) to (8) are added for dynamic point filtering, so the computational complexity O is:

$$O = O(P3P) + 8 \quad (9)$$

Fig. 7 The test results.

Comparison between the real trajectory and experimental trajectory. The blue line represents the real trajectory and the red line represents the experimental trajectory



Despite the increased complexity, it does not seriously affect the real-time performance and greatly improves the accuracy of the visual odometer for subsequent map construction.

4 Discussion

In this paper, we have developed an algorithm using spatial triangle constraints to restrict moving feature points in space. We verified that the triangle formed by three fixed points in space in different camera coordinates was close to similar. This method used ORB feature points for initial matching, and during the calculation of camera pose, we eliminated the dynamic points based on whether the triangles in the two camera coordinate systems were similar. We used two sets of image sequences on the TUM common dataset [27]. In the experiment, we extracted feature points from the RGB images and calculated the actual depth position of feature points with depth images. Finally, dynamic feature points were eliminated by structural constraints between feature points. Experimental results on the common dataset showed that the proposed approach reduces errors and effectively improves accuracy in the dynamic environment compared to the existing ORB point feature method. Therefore, the method proposed in this paper greatly decreases the effect of moving objects during camera pose estimation while also improving the accuracy and robustness of the visual odometer in dynamic environments.

Our method requires the 3D coordinates of the spatial points in the camera coordinate system. Due to the error of the Kinect camera itself, the depth information may be inaccurate or lost; thus, we need to re-estimate the depth of these feature points. Moreover, when selecting three sets of matched feature points, the dynamic points may be selected multiple times. These options increase computing time and reduce the running efficiency of SLAM. In the future, we could eliminate the dynamic points directly, rather than reselecting the initial points.

5 Conclusions

In order to improve the accuracy and robustness of visual SLAM in dynamic environments and to solve the problem of large deviations in pose estimation of visual SLAM systems due to the presence of moving objects in dynamic scenes, in this paper, we proposed a new visual odometry approach based on the structural relations between feature points in an image. This method used the spatial position information of feature points to determine whether the object is moving or not, and this approach could eliminate dynamic points when calculating camera pose. In the process of drawing construction, this method can get rid of the influence of dynamic objects in space, thus reducing the tracking error and improving the accuracy of drawing construction. We conducted our experiments on TUM public dynamic dataset [27]. The results show that the localization accuracy of our system is greatly improved compared to the traditional method in a dynamic environment.

In the future, we plan to add a semantic segmentation module to directly eliminate dynamic points and use the results of semantic segmentation to construct a semantic octree map, which improve the ability to avoid moving obstacles in dynamic scenes and it is useful for high-level robotic tasks.

Author contribution All named authors initially contributed a significant part to the paper. Gang Xu designed the study. Experimental model is built by Gang Xu and Ze Yu. Analyses were carried out by Gang Xu and Xingyu Zhang. Organization of data was led by Gang Xu and Guangxin Xing. Descriptions of text use were assisted by Ze Yu and Feng Pan.

Funding This work was sponsored by the Natural Science Foundation of Anhui Province of China (2108085MF197).

Availability of data and material The data used to support the findings of this study are included within the article.

Declarations

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

Conflict of interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Wang C, Cheng J, Wang J et al (2018) Efficient object search with belief road map using mobile robot. *IEEE Robot Autom Lett* 3(4):3081–3088
- Yang D, Bi S, Wang W et al (2019) DRESLAM: dynamic RGB-D encoder SLAM for a differential-drive robot. *Remote Sens* 11(4):380
- Yang S, Scherer SA, Yi X, Zell A (2017) Multi-camera visual SLAM for autonomous navigation of micro aerial vehicles. *Robot Auto Syst* 93:116–134
- Endres F, Hess J, Sturm J et al (2014) 3-D mapping with an RGB-D camera. *IEEE Trans Rob* 30(1):177–187
- Yin H, Ma Z, Zhong M et al (2020) SLAM-based self-calibration of a binocular stereo vision rig in real-time. *Sensors* 20(3):621
- Du ST, Li YF, Li XY, Wu MH (2021) LiDAR odometry and mapping based on semantic information for outdoor environment. *Remote Sens* 15(13):2864
- Bahraini MS, Bozorg M, Rad AB (2019) A new adaptive UKF algorithm to improve the accuracy of SLAM. *Int J Robot Theory Appl* 5:35–46
- Croce MD, Pire T, Bergero F (2018) DS-PTAM: distributed stereo parallel tracking and mapping SLAM system. *J Intell Robot Syst* 95(2):365–377
- Zhang S, Zheng L, Tao W (2021) Survey and evaluation of RGB-D SLAM. *IEEE Access* 9:21367–21387
- Campos C, Elvira R, Rodríguez JGG, Montiel J, Tardós JD (2021) ORB-SLAM3: an accurate open-source library for visual, visual-inertial, and multimap SLAM. *IEEE Trans Robot* 37(6):1874–1890
- Tetsu S, Michifumi Y, Katsufumi I (2020) Camera tracking improvement for LSD-SLAM system with 360-degree camera. *IEEJ Transactions on Electronics, Information and Systems* 140(7):800–809
- Akinlar C, Topal C (2011) EDlines: a real-time line segment detector with a false detection control. *Pattern Recogn Lett* 32(13):1633–1642
- Li C, Kang Z, Yang J, Li F, Wang Y (2020) Research on semantic-assisted SLAM in complex dynamic indoor environment. *Int Arch Photogramm Remote Sens Spat Inf Sci XLIII-B4–2020* 353–359
- Engel J, Koltun V, Cremers D (2018) Direct sparse odometry. *IEEE Trans Pattern Anal Mach Intell* 40(3):611–625
- Ban X, Wang H, Chen T, Wang Y, Xiao Y (2021) Monocular visual odometry based on depth and optical flow using deep learning. *IEEE Trans Instrum Meas* 70:1–19
- Costante G, Ciarfuglia TA (2017) LS-VO: learning dense optical subspace for robust visual odometry estimation. *IEEE Robot Autom Lett* 3(3):1735–1742
- Sun Y, Ming L, Meng QH (2017) Improving RGB-D SLAM in dynamic environments: a motion removal approach. *Robot Auton Syst* 89(Complete):110–122
- Wei H, & Zhang T, Zhang L (2021) GMSK-SLAM: a new RGB-D SLAM method with dynamic areas detection towards dynamic environments. *Multimed Tools Appl* 80(21–23):31729–31751
- Zou Y, Eldemiry A, Li Y, Chen W (2020) Robust RGB-D SLAM using point and line features for low textured scene. *Sensors* 20(17):4984
- Zhang C (2021) PL-GM:RGB-D SLAM with a novel 2D and 3D geometric constraint model of point and line features. *IEEE Access* 9:9958–9971
- Yang S, Scherer S (2017) Direct monocular odometry using points and lines. *IEEE Int Conf Robot Autom (ICRA) 2017*:3871–3877
- Kim DH, Kim JH (2016) Effective background model-based RGB-D dense visual odometry in a dynamic environment. *IEEE Trans Robot* 32(6):1565–1573
- Cheng J, Wang C, Mai X, Min Z, Meng QH (2021) Improving dense mapping for mobile robots in dynamic environments based on semantic information. *IEEE Sens J* 21(10):11740–11747
- Cheng J, Wang Z, Zhou H, Li L, Yao J (2020) DM-SLAM: a feature-based SLAM system for rigid dynamic scenes. *ISPRS Int J Geo Inf* 9(4):202
- Guo K, Ye H, Gu J, Chen H (2021) A novel method for intrinsic and extrinsic parameters estimation by solving perspective-three-point problem with known camera position. *Appl Sci* 11(13):6014
- Bansal M, Kumar M, Kumar M (2021) 2D object recognition: a comparative analysis of SIFT, SURF and ORB feature descriptors. *Multimed Tools Appl* 80(12):18839–18857
- Sturm J, Engelhard N, Endres F, Burgard W, Cremers D (2012) A benchmark for the evaluation of RGB-D SLAM systems. *IEEE/RSJ Int Conf Intell Robots Syst Vilamoura* 573–580

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.