International Journal of Trend in Research and Development, Volume 11(2), ISSN: 2394-9333
www.ijtrd.com

A Dense Map Construction Algorithm based on Motion Area Removal in Dynamic Environments

[1]Jun Dai, [1]Minghao Yang and [2]Naohiko Hanajima
[1]School of Mechanical and Power Engineering, Henan Polytechnic University, Jiaozuo, China
[2]Robotics and Mechanical Engineering Research Unit, Muroran Institute of Technology, Muroran, Japan

***Abstract*—**As one of the key technologies for robots to perceive real-world environments, the visual Simultaneous Localization And Mapping (SLAM) system utilizes only geometric spatial features in the process of mapping, which is unable to construct static dense maps in complex dynamic environments and is difficult to eliminate large quantities of drifting point clouds. To solve the problem, this paper proposes a map construction method based on a visual SLAM system and convolutional neural network. First, the dynamic area are comprehensively determined by object detection networks and geometric constraints, after which they are eliminated to remove the negative impact on pose estimation. Second, the dense building map is added to the back-end of our system, which combines the dynamic areas information to reject the drifting point cloud of motion, and then the camera pose and environment images are utilized to generate the dense map by stitching the point cloud together. Experimental results on public datasets show that our algorithm is able to construct accurate maps of dense static environments in dynamic environments, while effectively improving the SLAM system's localization accuracy and maintaining real-time capability of the system.

Keywords*—Visual SLAM; Pose estimation; Feature match; Object detection; Three-dimensional point cloud map;*

I. INTRODUCTION

Over recent years, benefiting from tremendous advances in artificial intelligence technology, intelligent robots are more widely applied. And as one of the key technologies for robots to autonomously localize and sense environmental information, the SLAM system is capable of estimating its position during its movement while constructing the map information of the surrounding environment. In traditional SLAM systems, map construction is mainly targeted at localization tasks, i.e., map information is used to assist in improving the localization accuracy of the system, which is usually in the form of sparse point cloud maps, which are unable to recognize semantic information in the environment, and cannot be applied to the needs of advanced tasks such as robot navigation, obstacle avoidance, and three-dimensional reconstruction. In addition, conventional SLAM systems are constructed relying on the assumption that static environments remain unchanged, yet the real world is always complicated and volatile. For example, people walking indoors, vehicles traveling on outdoor roads, and so on. The localization precisions of traditional SLAM systems in the above dynamic situations will be severely degraded, which will lead to the drift of the constructed maps and make them unusable for subsequent advanced tasks. Therefore, how to construct high-density environmental maps containing only static point clouds has become a critical problem to be solved. Due to the great progress of deep learning technology, many researchers have combined deep learning algorithms to construct semantic environment maps oriented to dynamic scenes, which mainly use convolutional neural networks to extract semantics from the images of the input system and fuse them with the 3D point cloud to exclude the drifting dynamic point cloud in order to construct a dense point cloud map that contains only static objects. McCorma et al. [1] proposed a fused semantic algorithm for environmental map construction using convolutional neural networks, which corrects the map in real-time by introducing semantic segmentation threads and optimizes the results of semantic segmentation based on the bit position information estimated between frames; however, such methods are too demanding in terms of computational resources, which leads to serious degradation of the system's real-time performance and makes it difficult to deploy the application practically on a robotic platform. In order to address this problem, Mao et al. [2] proposed a semantic map construction algorithm based on RTABMAP [3] and YOLO[4], which firstly uses YOLOv2 to detect the target on the color image to obtain the rough position information of the object, and then adopts the Canny edge detection method to refine the segmentation of the object region on the depth image, which effectively reduces the computational demand of the system and improves the SLAM system's real-time capability. Ehlers et al. [5] proposed a semantic map construction method for automatically adjusting map building parameters, which automatically adjusts the optimal parameters for building maps based on the results of semantic recognition, thus realizing the task of semantic map construction in arbitrary scenarios. However, in complex and changing dynamic environments, a single invariant parameter usually cannot meet the practical requirements. In view of this, many works have proposed semantic map construction methods oriented to dynamic environments using lightweight object detection. Hoang et al. [6] proposed a semantic map construction algorithm with strong robustness to address the problem of difficulty in estimating the accurate camera pose due to the negative influence of moving objects in dynamic environments, but since the system relies on the Elastic Fusion [7] algorithm, it is also constrained by computational resources, which leads to poor real-time capability of the system. Hosseinzadeh et al. [8] proposed a quick and accurate semantic mapping construction algorithm, which utilizes two convolutional neural networks for planar segmentation and parametric regression, respectively, and combines them with the object detection network for detection and tracking.

Although the above methods are able to accomplish the task of semantic map construction in static environments, however, in dynamic environments the maps constructed by these systems have a large number of drifting dynamic point clouds, and use object detection algorithms to extract the boundary position information of objects in the environment, there will be a problem that the pixels in the moving area contain other objects. For improvement of the above problems, we propose a dense map construction algorithm built on a lightweight object detection network and visual SLAM, which uses an improved non-maximum suppression combined with the object detection network to extract semantic information about real-world environment; and designs an initial dynamic point detection method based on feature points to address the

problem of negative interruptions with moving objects; and finally, carries out the real dynamic area in the back-end of the improved system of culling, and then use the accurate camera's pose information generated by the improved system to stitch to generate a global static dense point cloud map of the environment. Through experimental results on the public dynamic dataset indicate that our proposed system can effectively improve the localization accuracy, efficiently generate a global static dense 3D point cloud map of dynamic environment, and maintain real-time performance.

II. SYSTEM DESCRIPTION

A. System Overview

The proposed improved system of this paper is constructed based on ORB-SLAM3 [9] and the overall overview of our proposed system is illustrated in Fig. 1.

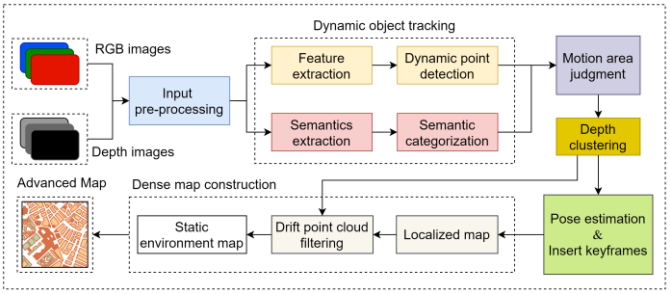


Fig. 1. Overview of the proposed system

First, ORB (Oriented FAST and Rotated BRIEF) feature points are extracted from the RGB image in the tracking thread of the visual odometry, and the sparse optical flow is utilized for feature matching and frame-to-frame pose estimation tracking, while embedding a lightweight object detection network YOLOv5 (The fifth version of You Only Look Once) to extract semantic information of objects in the environment. Second, the initial dynamic points present in the dynamic scenario are determined with standard geometric constraints. Last, a dense mapping thread is added to the back end of our system to generate an initial 3D point cloud from depth images and color images; then the motion regions in the environment are comprehensively determined, and the drifting point cloud in the real dynamic region is eliminated, and the semantic information of the keyframes is synchronously combined to stitch together to generate a static dense 3D point cloud map with global continuity and consistency. In addition, because the proposed system improves the effectiveness of short-term and medium-term data correlation, the system's localization accuracy in complex dynamic environments is greatly improved, especially in high dynamic environments, and maintains high system real-time performance.

B. Feature Matching based on Sparse Optical Flow

ORB feature points, as a kind of corner points with good scale invariance, are extensively used for the SLAM systems. However, in the process of feature extraction and matching in the front end of the visual SLAM, if the ORB feature points are extracted for all consecutive image frames, it will take up a large amount of computational resources, thus reducing the real-time performance of the whole system. Therefore, this paper proposes a feature tracking method based on LK (Lucas-Kanade, LK) optical flow to reduces the time-consumption of feature matching, which effectively reduce the computational consumption of feature extraction and matching, and improves the adaptability of the system to the weak texture degradation environment. Although the sparse optical flow is susceptible to changes in light intensity, since the Oriented FAST key points with good light invariance are used as feature points in this paper, the optimization is performed by fusion, which maintains the robustness of feature extraction and accelerates feature matching. The pixel points around the image feature points change over time and are called sparse optical flow. Obviously, if the image is set as a function of pixels with respect to time t, the grayscale of a pixel point can be expressed as $I(x, y, t)$. According to the assumption of grayscale invariance for the current moment t to the moment $t+dt$. Assuming that a pixel point moves to $(x+dx, y+dy)$, the gray level between adjacent frames can be expressed as:

$$I\left(x+dx, y+dy, t+dt\right) = I\left(x, y, t\right) \tag{1}$$

Then a firstorder Taylor expansion of the left-hand side of the above equation is obtained:

$$\begin{aligned} & I\left(x+dx, y+dy, t+dt\right) \\ & \approx I\left(x, y, t\right) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt \end{aligned} \tag{2}$$

Since the assumption of grayscale invariance assumes that the same spatial point gets pixel grayscale is invariant from position to position, it can be obtained:

$$\frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt = 0 \tag{3}$$

The above equation is divided by dt on both sides simultaneously to get:

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} = -\frac{\partial I}{\partial t} \tag{4}$$

The chi-square matrix is of the form:

$$\left[I_x, I_y\right] \times \begin{bmatrix} u \\ v \end{bmatrix} = -I_t \tag{5}$$

Where dx/dt denotes the velocity u of the pixel point in the x-axis; dy/dt is the velocity v of the pixel point in the y-axis; I_x and I_y are the gradient of the pixel in the x-direction and y-direction, respectively; and It is the bias of the pixel's gray level with respect to time. After that, the sliding window method is utilized to calculate the motion velocity of pixels for a fixed-size image block, assuming that a pixel block of size $s*s$ contains s^2 pixel points, which are approximated to have the same motion, and the overdetermined linear equation is constructed as:

$$\begin{bmatrix} I_x & I_y \end{bmatrix}_k = \begin{bmatrix} u \\ v \end{bmatrix} = -I_{t,k}, \quad k = 1, \ldots, s^2 \tag{6}$$

By solving the above super-definite equation, we can obtain the moving speeds of pixels between neighboring frames, so as to match the pixels of the previous frame to the relevant pixel positions in the current frame, and finally complete the feature matching between neighboring frames. In practical applications, multi-layer optical flow image pyramids are often combined to improve the feature mismatching problem. In our system, the amount of image pyramid layers is set to 6, and the scaling multiplier is set to 0.5.

C. Initial Dynamic Point Detection

For dynamic features in the environment, the judgment is mainly based on the extent to which they violate the geometric constraints in the static environment. Therefore, in this paper, the magnitude of the distance from the projection point of an image feature to the corresponding polar line is used as the basis for judging the extent to which the projection point

conforms to the pairwise pole constraints, and thus detecting abnormal dynamic outliers to be provided to the subsequent information fusion modules.

First, the feature matching point pairs between neighboring frames are obtained according to the improved FAST key point extraction and feature matching method described above, assuming that the matching feature point pairs are p_1 and p_2, respectively, and the chi-square coordinates of these image pixel points are:

$$p_1 = [u_1, v_1, 1]^T$$
$$p_2 = [u_2, v_2, 1]^T \quad (7)$$

Second, standard geometric constraints are constructed for the matched point pairs, and the standard pair of epipolar constraint equations are shown in the following equation:

$$p_2^T \times F \times p_1 = 0 \quad (8)$$

The Random Sample Consensus approach is utilized for solving the fundamental matrix of the interframe transformation and further filter the exact matching point pairs to compute again the exact fundamental matrix F. Then the matching point p_i in the previous frame in the neighboring frames is mapped to the corresponding polar line l in the current moment by the exact fundamental matrix F. Then the corresponding polar equation can be expressed as:

$$l_{i,j} = F \times p_i = F \times [u_i, v_i, 1]^T \quad (9)$$

Assuming that the linear equation coefficient vector of the polar line is (A_i, B_i, C_i), the equation of the polar line in the current frame is:

$$A_i x + B_i y + C_i = 0 \quad (10)$$

Where x, y denote the horizontal and vertical coordinates of the matching feature point in the corresponding pixel coordinate system, respectively. Finally, if one of the feature points obtained from the matching is distributed on a dynamic object, the standard geometric constraints will no longer be strictly satisfied. As a consequence, the feature point will not be located on the corresponding polar line, i.e., there is a certain distance deviation, then the pixel distance d between the dynamic feature point and its corresponding polar line can be calculated using the following equation:

$$d = \frac{|A_i x_i + B_i y_i + C_i|}{\sqrt{A_i^2 + B_i^2}} \quad (11)$$

If the pixel distance from the feature point to the polar line exceeds a specific threshold, the feature point can be considered a potential dynamic feature point; conversely, the feature point is considered a static feature point. In our system, the distance threshold is set to 1 pixel. In summary, the dynamic feature point detection method based on geometric constraints can effectively extract dynamic feature points within moving objects present in the environment without relying on accurate internal camera parameters. These rough initial dynamic feature point information will provide key posteriori constraints for the subsequent information fusion.

D. Semantic Extraction

Compared with traditional target detection algorithms, such as support vector machines, machine learning, and other solutions, deep learning-based object detection algorithms have the benefits of fast speed, high precision, and extensive recognition range. Nowadays, the excellent objectt detection frameworks can be mainly categorized into: Faster R-CNN [10], SSD [11], and YOLO algorithms. Among them, the YOLO series of algorithms has become one of the most representative algorithms in the field by virtue of its high efficiency and accuracy, at the same time, in order to meet the accuracy and real-time requirements of the SLAM system, this paper chooses the fifth version of the YOLO algorithm as the backbone network for semantic extraction, and its network structure is presented in Fig. 2.

As shown in the above figure, the network structure is a fully convolutional network containing only the convolutional kernel normalization layer and no fully connected layer. It is mainly composed of Backbone and Head. Firstly, features are extracted by attention mechanism and multi-layer convolution in the Backbone network, and more detailed and delicate features are obtained by combining C3 (three special convolutional layers) with downscaling and upscaling of the features, and finally the multi-scale features are fused using SPP (Spatial Pyramid Pooling, SPP). Next, this information is up-sampled and spliced at the Head, and by minimizing the loss error, the feature maps of classification probability, target bounding box, and target confidence are finally obtained, respectively. The loss error function expression of this algorithm is shown in the following equation:

$$Loss = a \times loss_{obj} + b \times loss_{rect} + c \times loss_{clc} \quad (12)$$

In the above equation, $loss_{obj}$, $loss_{rect}$, $loss_{clc}$ represent the confidence loss, bounding box loss, and classification loss, respectively; where a=0.4, b=0.3, and c=0.3 are the weight factors of the corresponding losses, respectively. In fact, the SLAM system focuses more on the obtained target bounding box information, so the proposed system of this paper focuses on improving the recall of the object detection algorithm to obtain more object bounding box as the prior semantics.

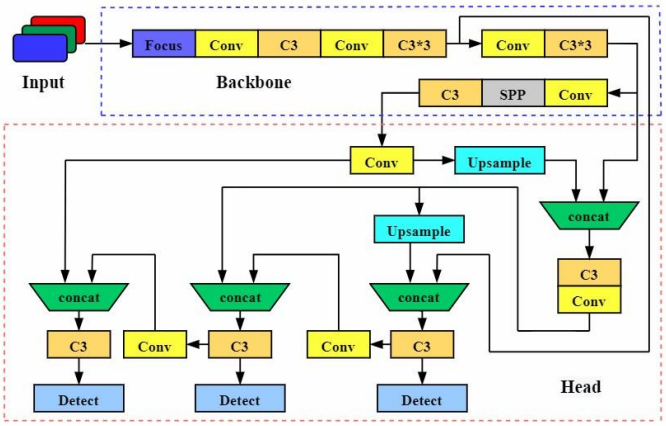


Fig. 2. Network structure of YOLOv5

In addition, it is worth noting that the model used in this paper was pre-trained on the MS COCO [12] dataset, hence our system is able to effectively recognize 80 object categories commonly found in the real world, such as persons, cars, bicycles, and other objects. According to the probability of these objects to undergo autonomous motion, we categorize them into high dynamic objects and low dynamic objects, which are classified as shown in Table 1, and different weights will be assigned to the dynamic objects with different motion attributes to participate in the subsequent fusion.

Table 1: Classification of properties of object motion

Motion Properties	Object Category
High dynamic	person, vehicles, animals, and 19 other objects
Low dynamic	desk, computers, traffic signs, and 61 other objects

E. Dynamic Area Rejection

In the proposed system in this paper, the core idea is to accurately eliminate the negative influence of dynamic features on the position estimation by fusing the semantic and geometric constraint information in the environment, so as to promote the system's localization accuracy under real dynamic environments, and to utilize the filtered dynamic regions to filter the drifting point cloud, and finally to generate a static dense global point cloud map. In fact, since the motion of objects is relative and dynamic objects usually do not keep a single motion state unchanged, different dynamic attributes are assigned to the objects, which in turn fuses the initial dynamic points and the dynamic object bounding box, and finally obtains the positional information of the real motion region. The specific process is shown in Fig. 3.

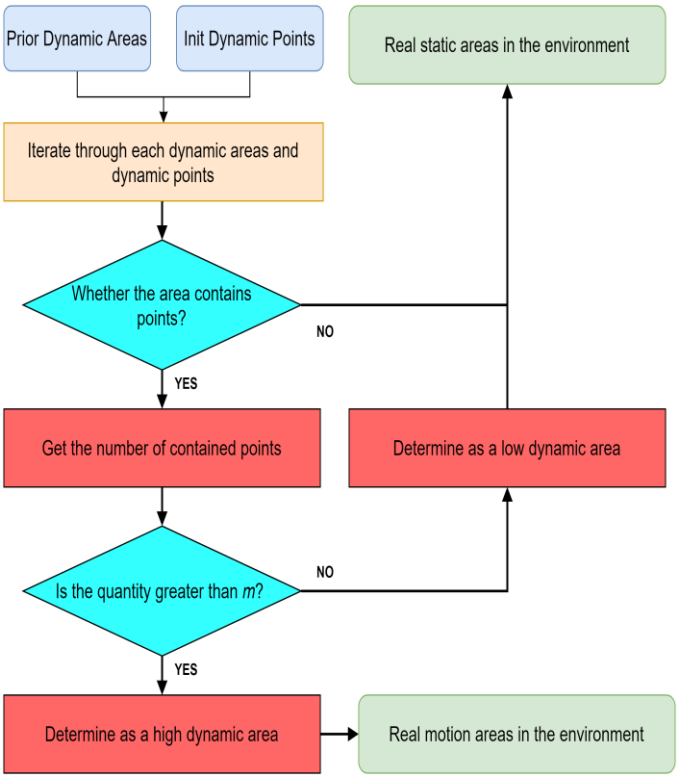


Fig. 3. The flow of real motion area judgment

When the real dynamic area is obtained according to the above method, the SLAM system will reject the feature points within the real dynamic region in the front end of the visual odometry to complete camera localization. Then, the accurate camera pose information and environment images are utilized to build the 3D dense map. Meanwhile, the dynamic area is fused in the mapping thread of our system, the drifting point cloud within the dynamic area is filtered, and a dense map of static environment is finally constructed.

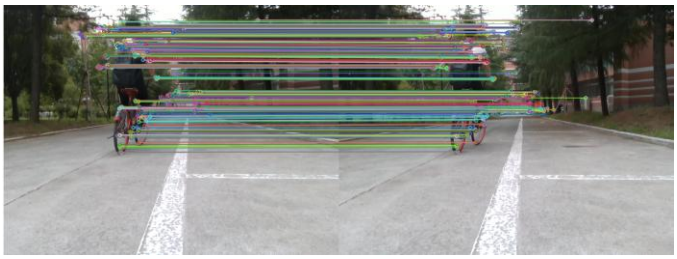
III. EXPERIMENTATION AND EVALUATION

To validate the effectiveness of the proposed system's localization and mapping within dynamic scenes, this paper experiments and evaluates the system's localization trajectory and map-building effect on the TUM [13] dynamic dataset. In addition, the dynamic point detection approach is tested in the dataset and the effect of dynamic areas judgment is demonstrated. The computing platform is a desktop computer, the software operating system is Ubuntu18.04, the CPU type is Intel i7-13700, and the RAM memory size is 32 GB. Without the requirement of GPU acceleration, the YOLOv5 network embedded in our SLAM system could achieve semantic extraction at about 20 FPS (Frames Per Second, FPS) while maintaining the real-time capability of our proposed system.

A. Effect of Dynamic Point Detection

The system was tested in the outdoor dynamic scenarios, where we tested the effectiveness of dynamic point detection and selected real dynamic points in neighboring image frames in the outdoor dynamic environments, respectively. Fig. 4 illustrates the effect of dynamic point detection.

(a) ORB feature point



(b) Sparse optical flow matching

(c) Dynamic feature point

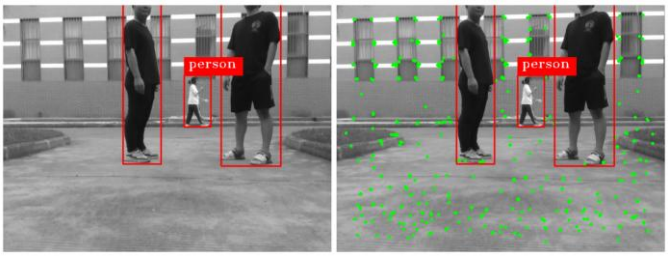
Fig. 4. The effect of dynamic point detection in the outdoor environment

In the above figure, the green points indicate the extracted feature points; the colored straight lines represent the feature-matching relationships between adjacent frames; the red points represent the initial dynamic points in the motion region; the blue points indicate the current initial static points. Obviously, our improved approach for feature matching does not show the phenomenon of lost tracking. Due to the combination of the improved FAST key point and optical flow matching methods, the robustness of tracking in weak texture environments can be maintained and the feature matching is accelerated. In conclusion, dynamic points within moving objects could be effectively detected in outdoor dynamic environments.

B. Effect of Motion Area Rejection

To demonstrate the effectiveness of the embedded object detection network, we conducts an experimental test of the semantic extraction module in real-world outdoor dynamic environment and fuses it with the initial dynamic points to judge the real motion region, and Fig. 5 specifically shows the effect of the real motion region judgment. The red dots in Fig. 5 denote the dynamic points and the red bounding box denotes the extracted the priori dynamic areas. As we can see from Fig. 5(d), a moving pedestrian is judged as a real motion area, and the dynamic feature points in this region are eliminated, while the feature points in the region of the standing person are effectively retained. Overall, our method achieves the accurate judgment for the real motion areas and effective rejection, which improve the accuracy of our system's pose.

(a) Raw RGB image (b) Initial dynamic point

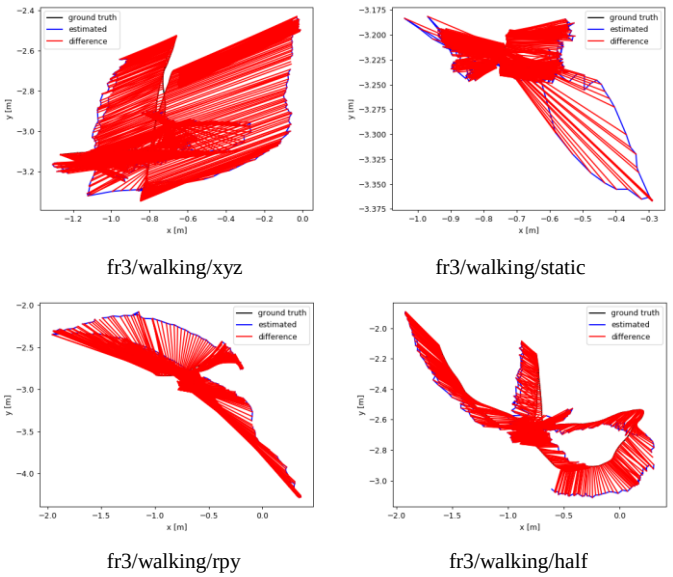


(c) Object bounding box (d) Retained static points

Fig. 5. The effect of dynamic feature point rejection

C. *Evaluation of Pose Estimation*

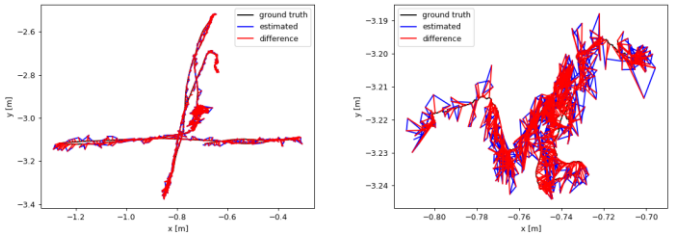
In this section, the fr3/walking sequence from the TUM dynamic dataset is selected for testing, which is categorized into four different camera motion modes, where xyz means that the camera is moving along the xyz coordinate axis; static means that the camera is manually kept stationary; rpy denotes that the camera is moving drastically along the rotational axis; and half means that the camera is moving along the surface of the hemisphere. These sequences contain a large number of dynamic objects that are a challenge for the localization of the SLAM system. We demonstrate the estimated trajectories of ORB-SLAM3 and our algorithm in Fig. 6 and 7, respectively. The blue line in Fig. 6 and 7 indicates the trajectory estimated by the SLAM system, and the black line is the ground truth trajectory captured by the external motion capture device. The specific localization error results are illustrated in Table 2.



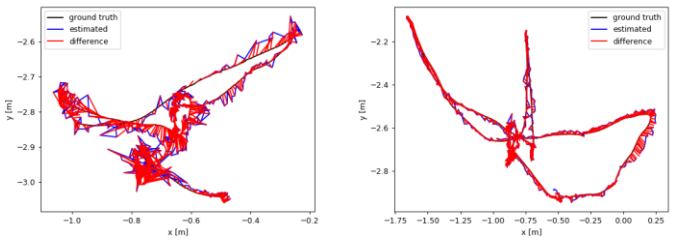
fr3/walking/xyz fr3/walking/static

fr3/walking/rpy fr3/walking/half

Fig. 6. The result of estimated trajectories for ORB-SLAM3



fr3/walking/xyz fr3/walking/static



fr3/walking/rpy fr3/walking/half

Fig. 7. The result of trajectories from our improved system

As shown in the above figure, the red portion visualizes the error distribution of the algorithm trajectories. Obviously, the localization accuracy of our proposed algorithm in this paper under dynamic environment sequences is higher than that of ORB-SLAM3, while the estimated trajectory of ORB-SLAM3 shows serious drift. The ATE (Absolute Trajectory Error, ATE) was used to quantitatively assess the error of the SLAM system and the performance was evaluated by RMSE. (Root Mean Square Error) and SD. (Standard Deviation).

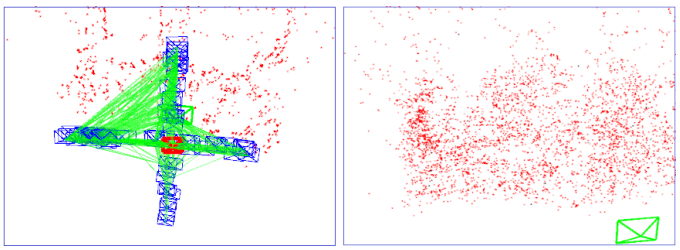
Table 2: The evaluation of ATE under dynamic scenario sequences

Sequences	ORB-SLAM3		Our algorithm	
	RMSE.	SD.	RMSE.	SD.
fr3/walking/xyz	0.5457m	0.2852m	0.0162m	0.0080m
fr3/walking/static	0.1719m	0.0695m	0.0067m	0.0029m
fr3/walking/rpy	0.6826m	0.3621m	0.0333m	0.0191m
fr3/walking/half	0.2489m	0.0781m	0.0268m	0.0133m

From the data analysis in Table 2, it can be observed that our proposed algorithm effectively improves the localization accuracy of the system in dynamic environments, and our system's RMSE in indoor dynamic scenarios is less than 3 cm. Compared to ORB-SLAM3, the proposed system improves the positional accuracy by more than 85% on average. As a whole, our algorithm is able to handle the interference of high number of moving objects on the dynamic scenes, improve the robustness and localization accuracy of SLAM system, and thus provide accurate pose information for the dense mapping module to stitch together the local point cloud.

D. *Effect of Dense Mapping*

To validate the effectiveness of our proposed algorithm for dense mapping in complex dynamic environments, the TUM dataset is selected for experimental testing and analyzed and evaluated in aspects of the map accuracy and computational complexity of the dynamic environment's geometric structure. The sequences selected for the experiment are fr3/walking/xyz, which contain a large number of static and dynamic objects. The sparse and dense environmental maps constructed by our improved system are shown in Fig. 8 and 9, respectively.



(a) system implementation (b) sparse point cloud

Fig. 8. Sparse maps constructed by the pre-improvement system

In Fig. 8, the original SLAM system can only generate sparse point cloud maps, which cannot visualize the geometric

texture information of the environment. Such maps are not suitable for tasks like robot navigation and obstacle avoidance.

(a) drift point clouds (b) static dense point clouds

Fig. 9. Dense maps constructed by our proposed system

Through comparing the maps generated by the original and improved systems, it is apparent that our proposed system is capable of constructing dense 3D point cloud maps. Fig. 9 (a) shows the drifting point cloud around a moving pedestrian, and by dynamic region rejection, the accuracy of the pose estimation is improved and the drifting point cloud of moving people is effectively eliminated, as indicated in Fig. 9 (b). It's worth noting that our proposed system utilizes only the image information from the keyframes to construct the 3D point cloud, and the system constructs the map at about 15 FPS. Overall, our proposed system is able to efficiently generate dense maps of static environments to generate static point cloud maps for future advanced application tasks.

CONCLUSION

Aiming at the problem that the classical visual SLAM system cannot construct a dense map of the static environment in real-world dynamic environment, a dense map construction method combining geometric constraints and deep learning networks is proposed on the basis of the conventional visual SLAM system. Our improved system mainly eliminates the negative interference of dynamic feature points for position estimation by means of dynamic region culling. For the dynamic drift point cloud existing in the dense mapping process, the drift point cloud is removed by combining with the real motion region judged by the system, and finally, a globally consistent static dense point cloud map is obtained. Extensive experimental results on the TUM dataset show that our proposed method can accurately and efficiently construct static environment dense maps in dynamic environments, and greatly improves the localization accuracy of the visual SLAM system, especially in high dynamic environments the system performs better, which verifies the validity of our proposed algorithm, and proves the robustness of our system.

References

[1] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "SemanticFusion: Dense 3D semantic mapping with convolutional neural networks," in 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, Singapore: IEEE, May 2017, pp. 4628–4635. doi: 10.1109/ICRA.2017.7989538.

[2] M. Mao, H. Zhang, S. Li, and B. Zhang, 'SEMANTIC-RTAB-MAP (SRM): A semantic SLAM system with CNNs on depth images', Math. Found. Comput., vol. 2, pp. 29–41, 2019.

[3] M. Labbe and F. Michaud, "Appearance-Based Loop Closure Detection for Online Large-Scale and Long-Term Operation," IEEE Trans. Robot., vol. 29, no. 3, pp. 734–745, Jun. 2013, doi: 10.1109/TRO.2013.2242375.

[4] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI: IEEE, Jul. 2017, pp. 6517–6525. doi: 10.1109/CVPR.2017.690.

[5] S. F. G. Ehlers, M. Stuede, K. Nuelle, and T. Ortmaier, 'Map Management Approach for SLAM in Large-Scale Indoor and Outdoor Areas', 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 9652–9658, 2020.

[6] D.-C. Hoang, T. Stoyanov, and A. J. Lilienthal, "Object-RPE: Dense 3D Reconstruction and Pose Estimation with Convolutional Neural Networks for Warehouse Robots," in 2019 European Conference on Mobile Robots (ECMR), Prague, Czech Republic: IEEE, Sep. 2019, pp. 1–6. doi: 10.1109/ECMR.2019.8870927.

[7] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison, 'ElasticFusion: Dense SLAM Without A Pose Graph', in Robotics: Science and Systems, 2015.

[8] M. Hosseinzadeh, K. Li, Y. Latif, and I. D. Reid, 'Real-Time Monocular Object-Model Aware Sparse SLAM', 2019 International Conference on Robotics and Automation (ICRA), pp. 7123–7129, 2018.

[9] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM," IEEE Trans. Robot., vol. 37, no. 6, pp. 1874–1890, Dec. 2021, doi: 10.1109/TRO.2021.3075644.

[10] S. Ren, K. He, R. Girshick, and J. Sun, 'Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks', IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, 2017.

[11] W. Liu et al., 'SSD: Single Shot MultiBox Detector', in Computer Vision -- ECCV 2016, 2016, pp. 21–37.

[12] T.-Y. Lin et al., 'Microsoft COCO: Common Objects in Context', in Computer Vision -- ECCV 2014, 2014, pp. 740–755.

[13] J. Sturm, N. Engelhard, F. Endres, W. Burgard and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 2012, pp. 573-580.