frontiers in Energy Research

ORIGINAL RESEARCH
published: 06 December 2021
doi: 10.3389/fenrg.2021.803631

# OC-SLAM: Steadily Tracking and Mapping in Dynamic Environments

*Zhenyu Wu*[1], *Xiangyu Deng*[2], *Shengming Li*[3] *and Yingshun Li*[4]*

[1]School of Innovation and Entrepreneurship of DUT, Dalian University of Technology, Dalian, China, [2]Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, China, [3]School of Computer Science and Technology, Dalian University of Technology, Dalian, China, [4]School of Control Science and Engineering, Dalian University of Technology, Dalian, China

Visual Simultaneous Localization and Mapping (SLAM) system is mainly used in real-time localization and mapping tasks of robots in various complex environments, while traditional monocular vision algorithms are struggling to cope with weak texture and dynamic scenes. To solve these problems, this work presents an object detection and clustering assisted SLAM algorithm (OC-SLAM), which adopts a faster object detection algorithm to add semantic information to the image and conducts geometrical constraint on the dynamic keypoints in the prediction box to optimize the camera pose. It also uses RGB-D camera to perform dense point cloud reconstruction with the dynamic objects rejected, and facilitates European clustering of dense point clouds to jointly eliminate dynamic features combining with object detection algorithm. Experiments in the TUM dataset indicate that OC-SLAM enhances the localization accuracy of the SLAM system in the dynamic environments compared with original algorithm and it has shown impressive performance in the localizition and can build a more precise dense point cloud map in dynamic scenes.

**Keywords: SLAM, dynamic environment, object detection, dense point cloud reconstruction, point cloud clustering**

OPEN ACCESS

***Edited by:***
*Xun Shen,*
*Tokyo University of Agriculture and Technology, Japan*

***Reviewed by:***
*Fengqiu Liu,*
*Ningbo University of Technology, China*
*Shi Zhang,*
*Northeastern University, China*

***Correspondence:**
*Yingshun Li*
*leeys@dlut.edu.cn*

***Specialty section:***
*This article was submitted to Smart Grids, a section of the journal Frontiers in Energy Research*

***Received:*** *28 October 2021*
***Accepted:*** *08 November 2021*
***Published:*** *06 December 2021*

***Citation:***
*Wu Z, Deng X, Li S and Li Y (2021) OC-SLAM: Steadily Tracking and Mapping in Dynamic Environments. Front. Energy Res. 9:803631. doi: 10.3389/fenrg.2021.803631*

## 1 INTRODUCTION

The indoor mobile robot is a robot system composed of multi-sensor fusion perception, autonomous decision making, mission planning, and control, etc. And from the perspective of the global mobile robot consumer market, its market scale is expanding, and various smart factories have great industrial demand for robots to complete various production tasks. For complex working environments, the first problem in autonomous mobile robots is the accuracy of localization and environmental map construction (Huang et al., 2019; Shen et al., 2020a). There has been a lot of outstanding work on SLAM research (Mur-Artal and Tardós, 2017; Engel et al., 2014; Qin et al., 2018), so we can build on these foundational frameworks to deal with tough issues.

In dynamic scenes, if the SLAM system fails to complete loop closure detection, the accuracy of pose estimation is seriously affected by dynamic features because the algorithm builds a map of the moving keypoints, resulting in poor system robustness and easily losing the tracking of camera pose. On the one hand, to solve these problems, some algorithms incorporate semantic segmentation or instance segmentation at the front-end of the visual odometry to obtain accurate edge information of moving objects, avoiding the influence of moving points from the feature extraction (Bescos et al., 2018; Kaneko et al., 2018; Runz et al., 2018; Yu et al., 2018; Zhong et al., 2018). Bescos *et al.* present a dynamic SLAM system based on ORBSLAM2 (Mur-Artal and Tardós, 2017) with Mask-RCNN semantic segmentation (Bescos et al., 2018), which contains monocular, binocular, and RGB-D inputs, and the extracted dynamic ORB features are rejected by invoking the Mask-RCNN model, but this

system is mainly time-consuming in the semantic segmentation algorithm and cannot achieve real-time pose estimation. Kaneko *et al.* present a monocular vision SLAM with a deep learning-based semantic segmentation method, using DeepLab v2 semantic segmentation of the mask to reject dynamic points and using CARLA simulator to provide new datasets for testing (Kaneko et al., 2018), but also faces the challenge of real-time. Runz *et al.* present RGBD-SLAM based on the aforementioned semantic segmentation and geometric segmentation, which can track dynamic objects and build corresponding 3D models that can be applied in AR (Runz et al., 2018). Yu *et al.* present a five threads dynamic SLAM system based on ORBSLAM2, adding a SegNet semantic segmentation thread and a semantic map thread to the original ORBSLAM2, and running in real-time with P4000 GPU (Yu et al., 2018). Doherty *et al.* build an IMU sensor based, semantic segmentation SLAM system which introduces data association into the SLAM system optimization process and performs land marker optimization, camera pose estimation and semantic information association simultaneously (Doherty et al., 2020). However, their approaches are fail to meet the demand for real-time operation and the single semantic segmentation algorithm does not guarantee the robustness of the SLAM system in the complex operating environment of the robot.

On the other hand, some notable results use the optical flow method for dynamic/static segmentation to highlight the dynamic semantics in the RGB images and provide the precise camera pose estimation and background reconstruction for robots (Alcantarilla et al., 2012; Jaimez et al., 2017; Zhang et al., 2020; Yu et al., 2021). Alcantara *et al.* present dense scene flow into visual SLAM, which performs scene flow calculation on images, and detects moving objects in the environment by comparing the scene flow changes of features (Alcantarilla et al., 2012), but the shortcomings of their method have been clearly recognized that time consumption severely affects the optical flow method, which is also restricted by the constant luminosity hypothesis. In addition to the aforementioned improvements to the front-end visual odometry, Henein *et al.* present a factor graph based back-end optimization method that incorporates moving point factors for dynamic objects to form constraints on feature observations, camera poses and dynamic object movement by semantic segmentation algorithms (Henein et al., 2020). Recently, some notable works focus their research on data association for dealing with the connection between semantic objects and RGB images in dynamic environments (Bowman et al., 2017; Doherty et al., 2019; Yu and Lee, 2018; Ran et al., 2021), and allow for better application of semantic techniques in SLAM algorithms. Furthermore, to deal with the uncertainty of environment, a potential approach is to improve SLAM algorithm by combining with various optimization-based algorithms (Wu and Shen, 2018; Shen et al., 2021; Shen et al., 2020b; Le et al., 2021; Wu et al., 2021; Toyoda and Wu, 2021) for scholastic systems.

Inspired by recent researches based on the semantic algorithm, we investigate the problem of real-time localization and dense map construction for the indoor mobile robots and propose a novel RGB-D SLAM framework which leverages a faster object detection method to obtain semantic information from RGB image and perform a dense map constuction with dynamic objects rejected.

Specifically, the main contributions of the SLAM framework presented in this paper are shown below:

- We design a real-time combined mismatch rejection algorithm based on the lightweight YOLO-Fastest object detection algorithm and Euclidean clustering method (OC-SLAM) where a robot can detect bad keypoints from dynamic objects through semantic information and point cloud clustering information. Especially, OC-SLAM is robust and computationally efficient in dynamic scenes.
- We present a dense point cloud reconstruction with dynamic objects rejected in OC-SLAM which leverages depth camera to directly obtain the depth image of scenes and remove dynamic objects in complex environments with Kd tree in order to create highly-precise dense maps.
- We evaluate OC-SLAM on a RGB-D benchmark dataset with the other state-of-the-art SLAM methods, and the proposed method achieves improved accuracy and robustness in dynamic scenes.

In the following section of this paper, we provide the framework of the proposed method OC-SLAM with the modules in the semantic object detection thread and dense mapping thread. Then **Section 3** includes experimental comparison with the original ORB-SLAM2 algorithm on TUM RGB-D dataset (Sturm et al., 2012). Ultimately, **Section 4** contains a brief discussion of the conclusions and results.

## 2 SYSTEM OVERVIEW

The dynamic objects in the robot operating environment will seriously affect the estimation of camera poses and mapping accuracy of the algorithm. Similarly, SLAM systems with monocular vision cameras cannot obtain real metric scale information in real complex environments. To accurately detect the dynamic features in the image, an improved algorithm is presented in this paper, whose overall framework is shown in **Figure 1**. Based on the original ORBSLAM2 (Mur-Artal and Tardós, 2017), a dense map reconstruction thread and an object detection thread are added in the system, and the identification of dynamic objects and the dense point cloud map reconstruction with dynamic objects removed is implemented by these two threads.

### 2.1 Dynamic Object Detection

You only look once (YOLO-Fastest) algorithm is now known to be the fastest and lightest improved version of the open-source YOLO universal object detection algorithm (Qiuqiu, 2021), which can run in real-time on the low-cost devices and consists of the convolutional neural network (CNN) (Long et al., 2015), so this paper utilizes the YOLO-Fastest detection algorithm and combines the geometric epipolar constraint

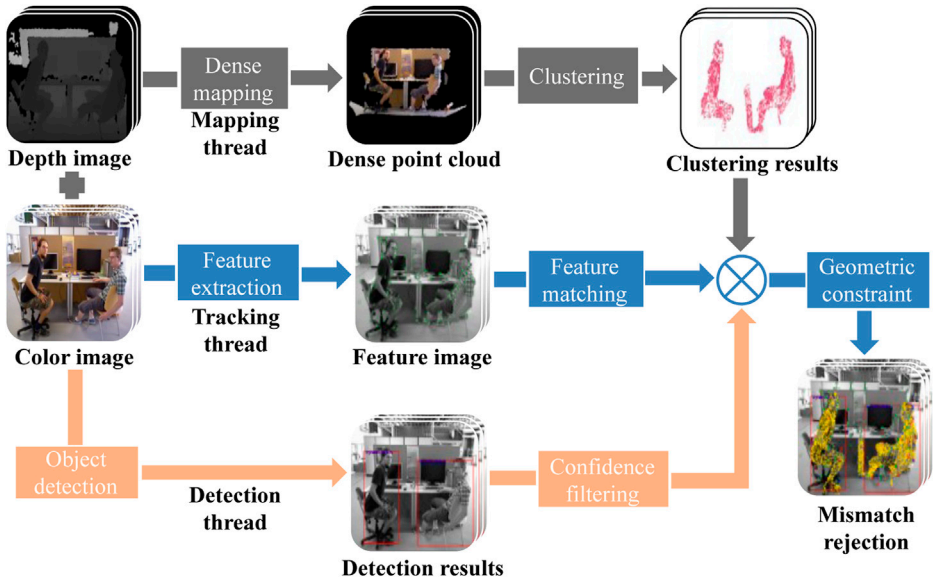


**FIGURE 1 |** The framework of the combined mismatch rejection algorithm, among which the tracking thread is as same as the original algorithm and the other two presented threads are added in the system.

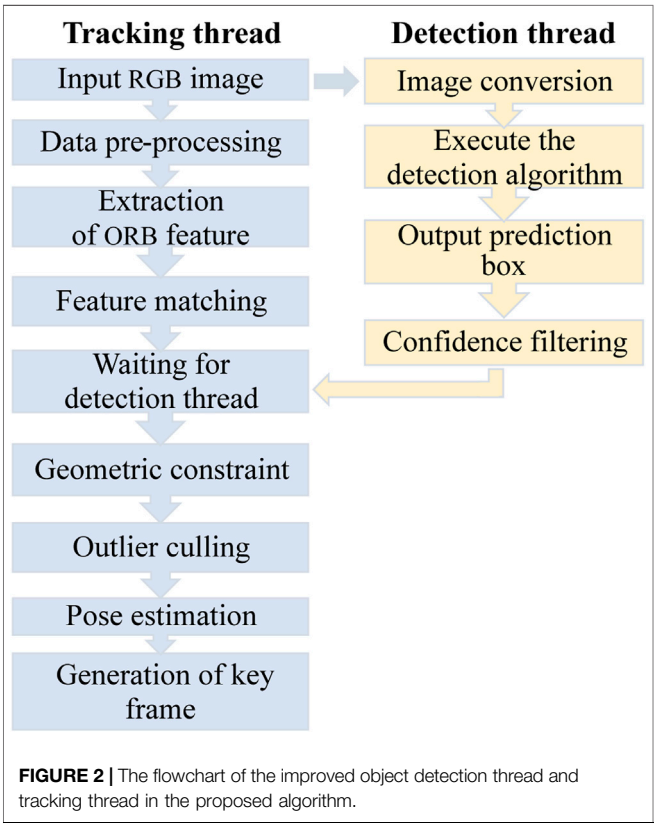


**FIGURE 2 |** The flowchart of the improved object detection thread and tracking thread in the proposed algorithm.

method for feature mismatch rejection, and further improves the original ORBSLAM2 system with three threads by adding the object detection thread for classification and localization of the original RGB image.

After the initialization of the SLAM system, the depth image is pre-processed to convert the depth map into real-scale depth data. As shown in **Figure 2**, The former thread is proposed to get the semantic information of the image and outputs the prediction box with confidence while the latter thread is improved to perform dynamic features rejection. The image is input to the YOLO algorithm for image detection after starting the object detection thread. While entering the main tracking thread, the extraction of image ORB features and the calculation of corresponding descriptors are started to complete the update of map points, and then the initial value of the camera pose is determined based on the working mode in which the main tracking thread is located, and the map points are reprojected and matched by the initial camera pose. The matching association between the map points and the current frame's features is discovered. When the system finishes feature matching, it exits the main tracking thread and waits for the YOLO object identification algorithm's detection result. Simultaneously, the prediction bounding box and confidence data are output by the object detection thread, where the results indicate the coordinates of the center point of a single prediction box, the width and height of the prediction box and the prediction confidence, and finally filter the information of the prediction boxes with confidence below 80, as shown in **Figure 3**, to obtain the prediction boxes of each target in the image.

## 2.2 Dynamic Geometrical Constraint

Therefore, when the object detection thread completes the image detection task, the matching feature pairs of the current frame are traversed within the main thread, and if the pixel coordinates of the features are within the prediction frame, the matching features outside the prediction frame are used to calculate the fundamental matrix $F$ of the current frame and the previous

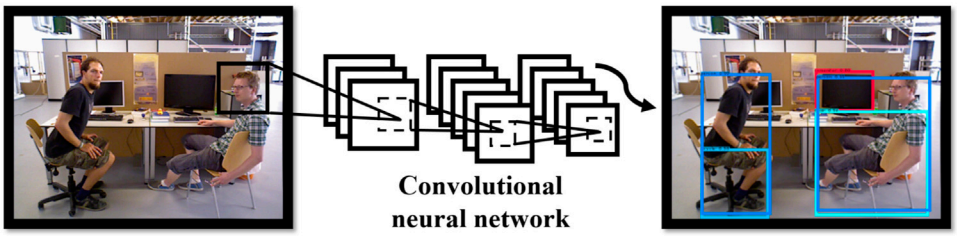


**FIGURE 3 |** The demonstration of YOLO-Fastest algorithm object detection for TUM dataset.

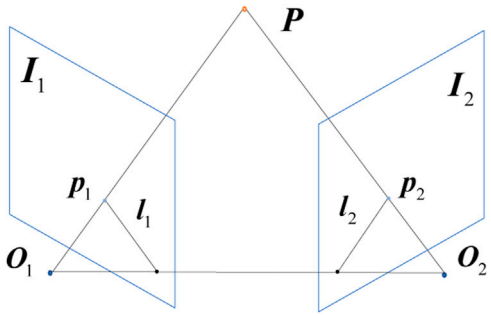


**FIGURE 4 |** The demonstration of epipolar geometry constraint.

images, and the distance from the reprojected epipolar lines to the corresponding matching features of the two adjacent frames is calculated by the method of geometric constraints (Andrew, 2001). If a point's distance error exceeds a threshold value set in a particular mode, the keypoint is considered an outlier, the corresponding map point matching association will be deleted. After the image feature extraction and matching process is completed, the camera pose estimation, local map establishment, and loop closure optimization process start implementation. As shown in **Figure 4**, $p_1$ and $p_2$ are the projection points of point $P$ on the two camera images $I_1$ and $I_2$, respectively, the point $p_1$ should be in the projection of the epipolar lines $l_1$ under ideal circumstances. As shown in **Eq. 1**, the calculation of the fundamental matrix $F$ between the current frame and the previous image can be defined as follows.

$$p_2^T F p_1 = 0, F = K^{-T} t \times R K^{-1}, \tag{1}$$

where $K$ is the intrinsic matrix, $t$ and $R$ are the translation and rotation matrix, respectively. As a result, the distance between the keypoint and the reprojection line may be computed using the fundamental matrix, as shown in **Eq. 2**:

$$d = \frac{p_2^T F p_1}{\sqrt{A^2 + B^2 + C^2}}, \tag{2}$$

where $d$ denotes the distance between points to lines, $A$, $B$ and $C$ denote the epipole line parameters. The minimum distance threshold is set based on the SLAM system's different modes (the distance threshold for the constant velocity motion model mode is smaller than the distance threshold for the keyframe mode), and if calculated distance exceeds threshold, the dynamic feature mismatch rejection is performed. Especially, the rejection of dynamic feature mismatch is not done when the SLAM system enters the relocalization mode because additional feature matching relationships are required for the initialization of the camera posture when the system enters the localization mode. Mismatch rejection is disabled in order to prevent the SLAM system from failing to initialize with insufficient features matching, which results in the loss of camera tracking. As shown in **Figure 5**, it depicts the result of dynamic feature rejection in the current frame

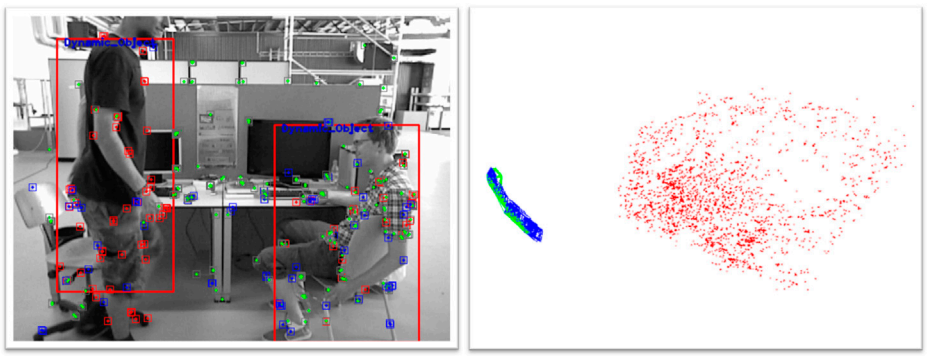

**FIGURE 5 |** Improved algorithm for current frame window with sparse point cloud map.

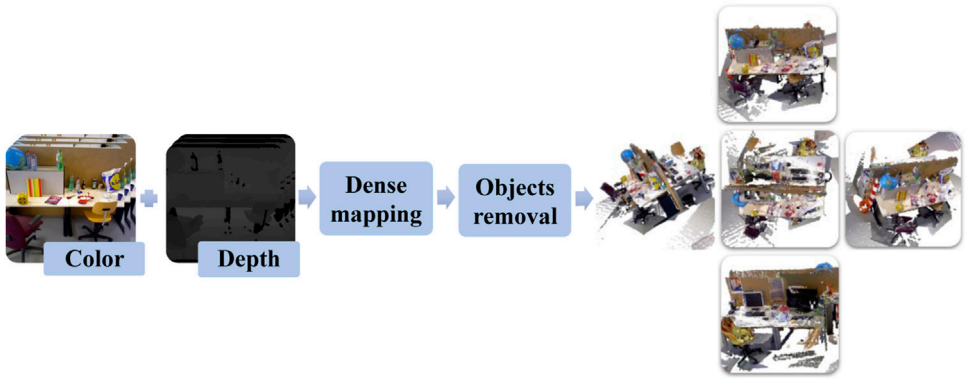


**FIGURE 6 |** The illustration of dense point cloud reconstruction thread in OC-SLAM system which is designed to perform the construction and clustering of dense point clouds without moving objects.

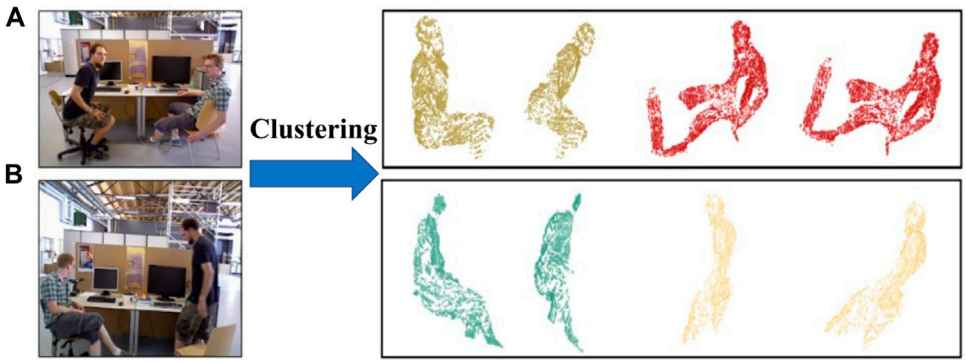


**FIGURE 7 |** Examples of Euclidean segmentation in TUM dataset, **(A)** is the point cloud segmentation with human bodies in the sitting posture, **(B)** are the clusters of the human body in the sitting and standing posture.

window with red dots indicating dynamic points and green dots indicating normal features, demonstrating that the enhanced method completes dynamic feature rejection properly. Moreover, the sparse point cloud generated from the features removes the map points from moving objects similarly in second image.

## 2.3 Dense Point Cloud Map Construction

Only sparse point cloud maps of features are built in the visualization thread of ORBSLAM2 system, which discards a large portion of the available map information. For this reason, sparse maps can not intuitively represent map information and are not available for other mission planning works such as navigation and obstacle avoidance by mobile robots that dense point cloud reconstruction is required. In this literature we introduce a new dense mapping thread to the ORBSLAM2 system, as shown in **Figure 6**, which is primarily utilized for dense point cloud reconstruction of the color and depth images Fernández-Madrigal (2012). If the coordinates of the picture sequence's points under the pixel coordinate $(\cdot)^P$ are $[u,v,1]^T$, then the coordinate values $[x,y,z]^T$ corresponding to those under the camera coordinate system $(\cdot)^C$ can be determined using **Eq. 3**:

$$\begin{cases} z = \dfrac{d}{s} \\ x = (u - c_x) \cdot \dfrac{z}{f_x} \\ y = (v - c_y) \cdot \dfrac{z}{f_y} \end{cases}, \tag{3}$$

where $d$ is the depth value of the image and $s$ is the depth metric scale of the camera. When the SLAM system inputs the depth map, its depth needs to be transformed to the real scale before it can be calculated. $c_x$, $c_y$, $f_x$ and $f_y$ are the camera intrinsic parameters. With the help of the camera extrinsic matrix, the pixel points can be converted from the coordinate system $(\cdot)^C$ to the real coordinates in the world coordinate system $(\cdot)^W$ as follows:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = (T \begin{bmatrix} x \\ y \\ z \end{bmatrix})_{1:3} = R \begin{bmatrix} x \\ y \\ z \end{bmatrix} + t, \tag{4}$$

where the coordinates $[X,Y,Z]^T$ represent the coordinate in the coordinate system $(\cdot)^W$, then the correspondence of points between the pixel coordinate and the world coordinate is

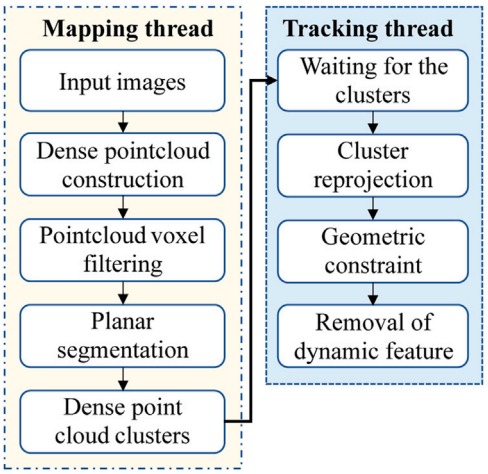


**FIGURE 8 |** The process of the rejection of dynamic objects. The presented mapping thread is illustrated in the yellow border which contains special operations on point clouds while the tracking thread is illustrated in the blue border which was improved to perform the task of dynamic keypoints rejection.

obtained, and the RGB value acquired from color image is set for each point cloud in the dense mapping thread, so that the basic dense point cloud is successfully constructed. However, in some practical applications, the pixel size of an image is usually 640 × 480, and the number of basic dense point clouds can be up to 300,000, so the point cloud voxel filtering and point cloud fusion are also needed for the basic point cloud.

## 2.4 Point Cloud Clustering Method

In this paper, the Euclidean Clustering method (Xiangyang et al., 2017) will be utilized to accomplish the point cloud segmentation task with the help of the YOLO-Fastest algorithm, which segments the point cloud data of the dense map into diverse single independent point cloud clusters. **Figure 7** illustrates the figures of two frames for 3D point cloud Euclidean clustering segmentation, when we input the depth image data from the dataset into the algorithm, a more accurate point cloud Euclidean segmentation result can be obtained with the assistance of semantic information from object detection method. Two point cloud clusters of human body in a sitting position with a well-defined point cloud profile extracted from the first image and the right corner of the table failed to remove through the filter since the human body is too close to the corner of table in Euclidean distance. In the second frame, a cluster of the human point cloud in sitting posture and a cluster of the human point cloud in standing posture are extracted, and the point cloud segmentation effect is better with no wrong clustering occurs.

## 2.5 Combined Mismatch Rejection Algorithm

The specified point cloud clusters in a frame are effectively separated after finishing the misson of Euclidean segmentation clustering of dense point cloud data. With this in mind, this paper presents a new mismatch rejection strategy algorithm for SLAM systems based on the Euclidean clustering method in OC-SLAM, which will be combined with an improved method based on the YOLO-Fastest object detection algorithm for jointly rejection of features of dynamic objects and ORB feature extraction in color image is carried out regularly on the main tracking thread, as shown in **Figure 8**. Moreover, feature matching is performed using different approaches depending on the incoming tracking mode and waits for the Euclidean clustering segmentation results in place once feature matching is accomplished. Accordingly, the dense mapping thread generates a sequence of independent point cloud clustering results by the use of the Euclidean clustering method, which includes point cloud dense reconstruction, voxel filtering and planar model segmentation. Afterwards, SLAM system set the dense build thread to idle. The tracking thread continues to implement after receiving the point cloud data from dense mapping thread, projecting each point cloud cluster into the pixel coordinate $(\cdot)^P$ using the equation:

$$\begin{cases} u = f_x \cdot x + c_x \\ v = f_y \cdot y + c_y \end{cases} . \tag{5}$$

The reprojection distance is calculated for the feature pairs contained in each point cloud according to the mismatching judgment method with respect to the epipolar constraint. Afterward, if more than half of the feature pairs fail to pass the geometrical constraint detection, the point cloud cluster is judged to be extracted from a moving object, and the features in the whole point cloud cluster and prediction box generated from the YOLO-Fastest algorithm are eliminated to perform the processing of moving objects removal in dynamic scenes.

## 2.6 Dynamic Object Rejection

Based on the previous work, the dense point cloud map is refined further and the clustered point cloud clusters of moving object in the base dense point cloud map are eliminated by constructing a Kd-Tree based on the results of point cloud Euclidean clustering, resulting in an environment map devoid of dynamic objects. As shown in **Figure 9**, two sets of color maps and depth images are input for dense construction: **Figure 9A** is the original color image, **Figure 9B** is the result of dense point cloud reconstruction and **Figure 9C** is the dense point cloud map with dynamic objects removed in which the point cloud clusters belonging to moving objects are essentially removed using the Euclidean clustering algorithm.

# 3 EXPERIMENTALS AND RESULTS

In this section, the improved algorithm is tested and validated on the TUM dataset from the Technical University of Munich, which collects image data in different experimental environments using Microsoft's Kinect camera and provides camera trajectory groundtruth for each dataset to evaluate the accuracy of the SLAM algorithm. In this research, dynamic and static

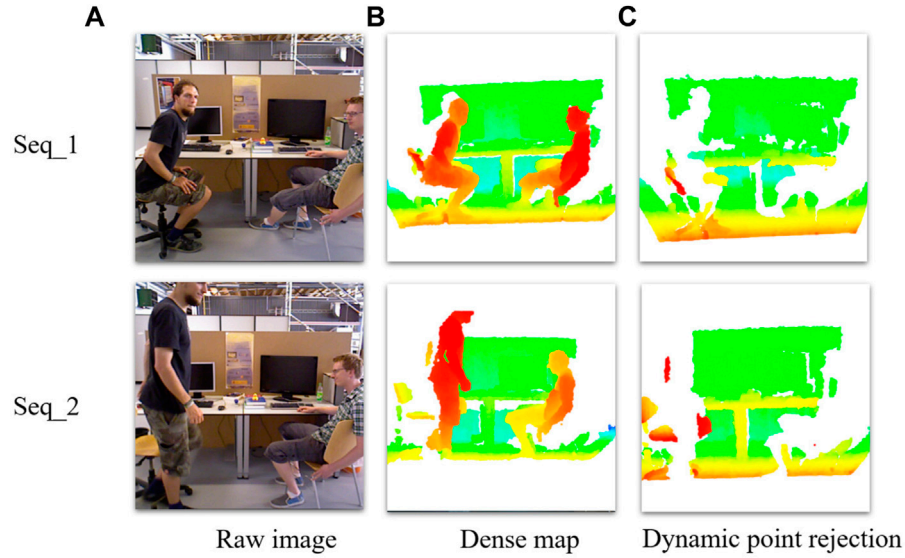


**FIGURE 9 | (A)** are the raw images, **(B)** are the point cloud dense reconstruction, (C) are the dense reconstruction with dynamic object rejection from which it can be seen that the clusters of moving human body in the dataset are removed by the improved algorithm.

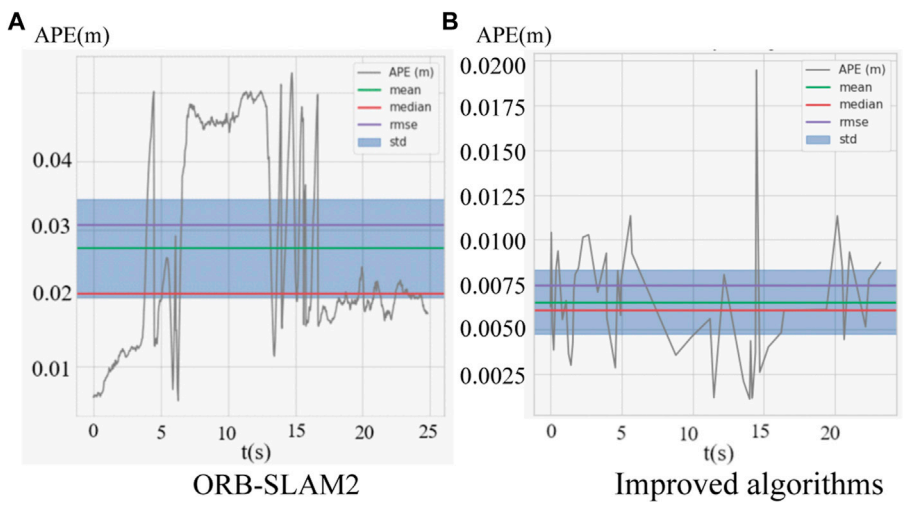


**FIGURE 10 | (A)** is the absolute trajectory error distribution of the combined improved algorithm, **(B)** is the absolute trajectory error distribution of the ORBSLAM2.

environment data are utilized to test the enhanced algorithm's accuracy of camera pose estimation and dense map construction performance.

## 3.1 Trajectory Estimation Experiments

In order to verify the robustness and accuracy of the improved algorithm's pose estimation, experiments under different complex environments are designed in this paper. The Root Mean Squared Error (RMSE) is used as the evaluation criterion for the absolute camera trajectory error (Sturm et al., 2012), and the RMSE of the estimated poses at all moments is calculated as follows:

$$RMSE(E) = \frac{1}{n}\sum_{t=1}^{n} \|trans(E_t)\|^2, \tag{6}$$

where error $E_t$ denotes the absolute trajectory estimation error (ATE) of the SLAM system at moment $t$, which is obtained by the calculation of the difference between the estimated trajectory of the camera pose and the groundtruth of the dataset. $trans(\cdot)$ indicates the translation of absolute trajectory estimation error $E_t$ and the enhancement effect in the experiment is calculated as the relative enhancement rate of the combined improved algorithm trajectory error with respect to the original algorithm. As shown in **Figure 10**, **Figure 10A** is the absolute trajectory error graph of the

**TABLE 1 |** The comparison of absolute trajectory error of pose estimation in TUM dataset.

| Image sequence | ORB-SLAM2(m) | | | Proposed(m) | | | Improvements(%) | | |
|---|---|---|---|---|---|---|---|---|---|
| | RMSE | Mean | Media | RMSE | Mean | Media | RMSE | Mean | Media |
| walking_static | 0.325 | 0.284 | 0.213 | 0.007 | 0.006 | 0.006 | 97.8 | 97.8 | 97,1 |
| walking_xyz | 0.756 | 0.655 | 0.653 | 0.129 | 0.119 | 0.118 | 84.1 | 81.7 | 81.9 |
| walking_half | 0.426 | 0.433 | 0.414 | 0.083 | 0.085 | 0.080 | 80.4 | 80.3 | 80.6 |
| sitting_static | 0.008 | 0.008 | 0.007 | 0.008 | 0.008 | 0.007 | −1.1 | −3.7 | 2.6 |

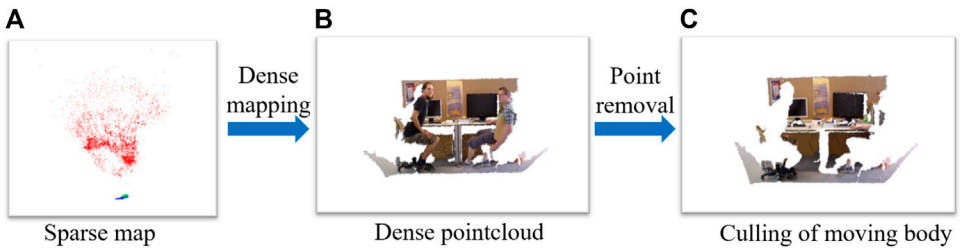


**FIGURE 11 |** The illustration of presented algorithm for dense point cloud map construction performance in the normal dataset, **(A)** is the sparse map, **(B)** is the dense point cloud map construction result, **(C)** is the dense map with moving body rejected.


**FIGURE 12 |** The illustration of presented algorithm for dense point cloud map construction performance in the dynamic dataset, **(A)** is the sparse map, **(B)** is the dense point cloud map construction result.

original algorithm without Loop closure and **Figure 10B** is the error evaluation graph of the combined algorithm, it can be seen that the majority of the time the error is below 0.01 m in the improved algorithm except for some extreme cases. Notably, at the moment of object detection algorithm failure, the Euclidean clustering module can continue to carry out the rejection of mismatch, which complements the object detection module to increase the robustness of the system and reduces the overall trajectory absolute error. Likewise, indicators of the median and mean trajectory error have significantly improved. Further, the error comparison between the improved algorithm and the original algorithm is shown in **Table 1**. And the evaluation indexes of the improved algorithm in the dynamic data sequences walking_static, walking_xyz, walking_half without loop closure are better than the original algorithm while the accuracy improvement effect is up to 97.8%, In spite of this, the accuracy in the image sequences in the static environment is approximately equal to that of the original algorithm in the static environment, indicating that the improvement modules in the algorithm do not lose too much algorithm performance. Importantly, the processing time per frame is only 97 ms on a low-performance processor, while the DynaSLAM (Bescos et al., 2018) algorithm takes 195 ms for the Mask R-CNN module alone using the Nvidia Tesla M40 GPU. Therefore, compared with the improved method using Mask R-CNN, the improved algorithm in this paper greatly improves the operation speed of the algorithm without excessive loss of accuracy.

## 3.2 Dense Reconstruction Experiment

Based on the successful detection and recognition of dynamic point cloud clusters, this paper performs point cloud dense building experiments on the improved algorithm, inputting normal image sequences in TUM dataset and image sequences in dynamic scenes to compare the dense building performance of the improved algorithm in two different dataset environments. As shown in **Figure 11**, in the

dense reconstruction experiment under the normal environment dataset, **Figure 11A** shows the sparse point cloud map established by the original system where the red points represent the map points successfully observed and the black points represent the map points observed in the current frame. Since the algorithm only calculates map points from the extracted features and performs fusion operation for redundant map points, only the sparse point cloud map is established. **Figures 11B,C** show the dense point cloud map built by the improved algorithm, which completely recovers the point cloud data in the dataset and further extracts more image information from the image sequence, making the mapping performance of the SLAM system more intuitive and the normal line of the map can be further calculated subsequently, thus reconstructing the network from the point cloud and converting the point cloud into a grid map. By contrast, as shown in **Figure 12**, in the dense reconstruction experiments under dynamic scene datasets, **Figure 12A** shows the sparse point cloud map built by the original system, which is built with low accuracy and fluctuating map updating with wrong map points due to the influence brought by fast-moving dynamic objects, thus leading to poor back-end nonlinear optimization of camera poses and map points. With this in mind, **Figure 12B** shows the dense point cloud map built by the improved algorithm, which not only recovers the specific scenes in the dataset completely but also uses the YOLO-Fastest object detection algorithm and the Euclidean clustering algorithm to eliminate the dynamic objects clusters in the dynamic scenes and retains the information of static objects in the point cloud map, which improves the robustness and accuracy of the dense point cloud mapping.

## 4 CONCLUSION

In this paper, we present an improved semantic SLAM algorithm (OC-SLAM) based on YOLO-Fastest object detection and Euclidean clustering method to reduce the impact of dynamic features on the accuracy of camera trajectory calculation by special processing of tricky issues in dynamic scenes to solve the problem of pose estimation and dense map construction. In comparison to Mask R-CNN and other semantic segmentation recognition methods, the proposed algorithm in this paper can greatly accelerate computation speed by leveraging the characteristics of the YOLO-Fastest algorithm to meet the algorithm's real-time requirements without sacrificing pose estimation accuracy. The absolute trajectory error(ATE) experiments in the TUM dataset indicate that this approach can increase accuracy on a low-performance embedded devices and build a dense point cloud map in the complex environment with dynamic objects eliminated.

## DATA AVAILABILITY STATEMENT

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

## ETHICS STATEMENT

Written informed consent was obtained from the individual(s) for the publication of any potentially identifiable images or data included in this article.

## AUTHOR CONTRIBUTIONS

ZW and XD contributed to the conception or design of the work; or the acquisition, analysis or interpretation of data for the work; XD drafted the work or revised it critically for important intellectual content; SL provided approval for publication of the content; YL agrees to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

## FUNDING

This work was financially supported by the Department of Science and Technology of Liaoning Province on "Research on the key technology of distributed energy networking based on wireless energy transfer" (Grand No. ZX20180613).

## SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fenrg.2021.803631/full#supplementary-material

## REFERENCES

Alcantarilla, P. F., Yebes, J. J., Almazan, J., and Bergasa, L. M. (2012). "On Combining Visual Slam and Dense Scene Flow to Increase the Robustness of Localization and Mapping in Dynamic Environments," in 2012 IEEE International Conference on Robotics and Automation, 1290–1297. doi:10.1109/ICRA.2012.6224690

Andrew, A. (2001). Multiple View Geometry in Computer Vision. *Kybernetes.* 30, 1333. doi:10.1108/k.2001.30.9_10.1333.2

Bescos, B., Facil, J. M., Civera, J., and Neira, J. (2018). Dynaslam: Tracking, Mapping, and Inpainting in Dynamic Scenes. *IEEE Robot. Autom. Lett.* 3, 4076–4083. doi:10.1109/LRA.2018.2860039

Bowman, S. L., Atanasov, N., Daniilidis, K., and Pappas, G. J. (2017). "Probabilistic Data Association for Semantic Slam," in 2017 IEEE International Conference on Robotics and Automation (ICRA), 1722–1729. doi:10.1109/ICRA.2017.7989203

Doherty, K., Fourie, D., and Leonard, J. (2019). "Multimodal Semantic Slam With Probabilistic Data Association," in 2019 International Conference on Robotics and Automation, Montreal, QC, Canada (ICRA), 2419–2425. doi:10.1109/ICRA.2019.8794244

Doherty, K. J., Baxter, D. P., Schneeweiss, E., and Leonard, J. J. (2020). "Probabilistic Data Association via Mixture Models for Robust Semantic Slam," in 2020 IEEE International Conference on Robotics and Automation (ICRA), 1432–1882. doi:10.1109/icra40945.2020.9197382

Engel, J., Schöps, T., and Cremers, D. (2014). "Lsd-slam: Large-Scale Direct Monocular Slam," in *Computer Vision – ECCV 2014*. Editors D. Fleet,

T. Pajdla, B. Schiele, and T. Tuytelaars (Cham: Springer International Publishing), 834–849. doi:10.1007/978-3-319-10605-2_54

Fernández-Madrigal, J.-A. (2012). *Simultaneous Localization and Mapping for Mobile Robots: Introduction and Methods: Introduction and Methods (IGI Global), Hershey, Pennsylvania, USA, IGI Global*.

Henein, M., Zhang, J., Mahony, R., and Ila, V. (2020). "Dynamic Slam: The Need for Speed," in 2020 IEEE International Conference on Robotics and Automation (ICRA), 2123–2129. doi:10.1109/ICRA40945.2020.9196895

Huang, B., Zhao, J., and Liu, J. (2019). A Survey of Simultaneous Localization and Mapping. arXiv preprint arXiv:1909.05214.

Jaimez, M., Kerl, C., Gonzalez-Jimenez, J., and Cremers, D. (2017). "Fast Odometry and Scene Flow from Rgb-D Cameras Based on Geometric Clustering," in 2017 IEEE International Conference on Robotics and Automation (ICRA), 3992–3999. doi:10.1109/ICRA.2017.7989459

Kaneko, M., Iwami, K., Ogawa, T., Yamasaki, T., and Aizawa, K. (2018). "Mask-slam: Robust Feature-Based Monocular Slam by Masking Using Semantic Segmentation," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 371–3718. doi:10.1109/CVPRW.2018.00063

Le, S., Wu, Y., Guo, Y., and Del Vecchio, C. (2021). "Game Theoretic Approach for a Service Function Chain Routing in Nfv with Coupled Constraints," in IEEE Transactions on Circuits and Systems II: Express Briefs, 1. doi:10.1109/TCSII.2021.3070025

Long, J., Shelhamer, E., and Darrell, T. (2015). "Fully Convolutional Networks for Semantic Segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 3431–3440. doi:10.1109/cvpr.2015.7298965

Mur-Artal, R., and Tardós, J. D. (2017). Orb-slam2: An Open-Source Slam System for Monocular, Stereo, and Rgb-D Cameras. *IEEE Trans. Robot.* 33, 1255–1262. doi:10.1109/TRO.2017.2705103

Qin, T., Li, P., and Shen, S. (2018). Vins-mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Trans. Robot.* 34, 1004–1020. doi:10.1109/TRO.2018.2853729

[Dataset] Qiuqiu, D. (2021). Yolo-fastest: yolo-fastest-v1.1.0. Available at: https://github.com/dog-qiuqiu/Yolo-Fastest.

Ran, T., Yuan, L., Zhang, J., He, L., Huang, R., and Mei, J. (2021). Not only Look but Infer: Multiple Hypothesis Clustering of Data Association Inference for Semantic Slam. *IEEE Trans. Instrum. Meas.* 70, 1–9. doi:10.1109/TIM.2021.3074954

Runz, M., Buffier, M., and Agapito, L. (2018). "Maskfusion: Real-Time Recognition, Tracking and Reconstruction of Multiple Moving Objects," in 2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 10–20. doi:10.1109/ISMAR.2018.00024

Shen, X., Ouyang, T., Khajorntraidet, C., Li, Y., Li, S., and Zhuang, J. (2021). Mixture Density Networks-Based Knock Simulator. *Ieee/asme Trans. Mechatron.*, 10. doi:10.1109/TMECH.2021.3059775

Shen, X., Zhang, X., Ouyang, T., Li, Y., and Raksincharoensak, P. (2020a). Cooperative Comfortable-Driving at Signalized Intersections for Connected and Automated Vehicles. *IEEE Robot. Autom. Lett.* 5, 6247–6254. doi:10.1109/LRA.2020.3014010

Shen, X., Zhang, Y., Sata, K., and Shen, T. (2020b). Gaussian Mixture Model Clustering-Based Knock Threshold Learning in Automotive Engines. *Ieee/asme Trans. Mechatron.* 25, 2981–2991. doi:10.1109/TMECH.2020.3000732

Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. (2012). "A Benchmark for the Evaluation of Rgb-D Slam Systems," in 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 573–580. doi:10.1109/IROS.2012.6385773

Toyoda, M., and Wu, Y. (2021). Mayer-type Optimal Control of Probabilistic Boolean Control Network With Uncertain Selection Probabilities. *IEEE Trans. Cybern.* 51, 3079–3092. doi:10.1109/TCYB.2019.2954849

Wu, Y., Guo, Y., and Toyoda, M. (2021). Policy Iteration Approach to the Infinite Horizon Average Optimal Control of Probabilistic Boolean Networks. *IEEE Trans. Neural Netw. Learn. Syst.* 32, 2910–2924. doi:10.1109/TNNLS.2020.3008960

Wu, Y., and Shen, T. (2018). A Finite Convergence Criterion for the Discounted Optimal Control of Stochastic Logical Networks. *IEEE Trans. Automat. Contr.* 63, 262–268. doi:10.1109/TAC.2017.2720730

Xiangyang, C., Yang, Y., and Yunfei, X. (2017). Measurement of Point Cloud Data Segmentation Based on Euclidean Clustering Algorithm. *Bull. Surv. Mapp.* 0, 27–31. doi:10.13474/j.cnki.11-2246.2017.0342

Yu, C., Liu, Z., Liu, X.-J., Xie, F., Yang, Y., Wei, Q., et al. (2018). "Ds-Slam: A Semantic Visual Slam Towards Dynamic Environments," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1168–1174. doi:10.1109/IROS.2018.8593691

Yu, H. W., and Lee, B. H. (2018). "A Variational Feature Encoding Method of 3d Object for Probabilistic Semantic Slam," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 3605–3612. doi:10.1109/IROS.2018.8593831

Yu, X., Wu, Y., Sun, X.-M., and Zhou, W. (2021). A Memory-Greedy Policy With Guaranteed Convergence for Accelerating Reinforcement Learning. *J. Autonomous Vehicles Syst.* 1, 011005–011012. doi:10.1115/1.4049539

Zhang, T., Zhang, H., Li, Y., Nakamura, Y., and Zhang, L. (2020). "Flowfusion: Dynamic Dense Rgb-D Slam Based on Optical Flow," in 2020 IEEE International Conference on Robotics and Automation (ICRA), 7322–7328. doi:10.1109/ICRA40945.2020.9197349

Zhong, F., Wang, S., Zhang, Z., Chen, C., and Wang, Y. (2018). "Detect-Slam: Making Object Detection and Slam Mutually Beneficial," in 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 1001–1010. doi:10.1109/WACV.2018.00115

**Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.