Journal of Intelligent & Robotic Systems (2024) 110:27
https://doi.org/10.1007/s10846-023-02046-3

REGULAR PAPER

RGB-D Based Visual SLAM Algorithm for Indoor Crowd Environment

Jianfeng Li[1,2,3] · Juan Dai[1,2,3] · Zhong Su[1,2,3] · Cui Zhu[4]

Received: 10 April 2023 / Accepted: 25 December 2023 / Published online: 2 February 2024
This is a U.S. Government work and not under copyright protection in the US; foreign copyright protection may apply 2024

Abstract
Most current research on dynamic visual Simultaneous Localization and Mapping (SLAM) systems focuses on scenes where static objects occupy most of the environment. However, in densely populated indoor environments, the movement of the crowd can lead to the loss of feature information, thereby diminishing the system's robustness and accuracy. This paper proposes a visual SLAM algorithm for dense crowd environments based on a combination of the ORB-SLAM2 framework and RGB-D cameras. Firstly, we introduced a dedicated target detection network thread and improved the performance of the target detection network, enhancing its detection coverage in crowded environments, resulting in a 41.5% increase in average accuracy. Additionally, we found that some feature points other than humans in the detection box were mistakenly deleted. Therefore, we proposed an algorithm based on standard deviation fitting to effectively filter out the features. Finally, our system is evaluated on the TUM and Bonn RGB-D dynamic datasets and compared with ORB-SLAM2 and other state-of-the-art visual dynamic SLAM methods. The results indicate that our system's pose estimation error is reduced by at least 93.60% and 97.11% compared to ORB-SLAM2 in high dynamic environments and the Bonn RGB-D dynamic dataset, respectively. Our method demonstrates comparable performance compared to other recent visual dynamic SLAM methods.

Keywords Visual SLAM · Indoor environment · Object detection · Dynamic environment

1 Introduction

With the advancement of technology, robots are gradually finding applications in large indoor environments like shopping malls and warehouses. SLAM technology plays a crucial role in enabling these machines to navigate and understand their surroundings. In unknown and unstructured environments, SLAM serves as a prerequisite technology, involving the creation of environmental maps and the estimation of the robot's pose within these maps.

Present-day research on SLAM primarily operates under the assumption of static scenes, as exemplified by systems like ORB-SLAM2 and ORB-SLAM3 [1, 2]. However, the presence of dynamic objects can lead to erroneous assessments of robots. When dynamic objects appear in the scene, it significantly affects the image frames, disrupting the initial feature point matching. This disruption results in the loss of feature tracking in visual odometry and substantial deviations in pose estimation. Furthermore, it also affects the reconstruction of maps. Researchers have proposed various methods to address this challenge [3–7].

Yu et al. [7] introduced an RGB-D-based DS-SLAM system that incorporates a semantic segmentation network, SegNet [8], to preprocess RGB images prior to map construction. This system effectively segments dynamic objects in the scene. Nevertheless, due to its reliance on direct image intensity information, it may encounter limitations in scenarios

✉ Juan Dai
daijuan@bistu.edu.cn

Jianfeng Li
2740470909@qq.com

Zhong Su
sz@bistu.edu.cn

Cui Zhu
cuizhu_lzy@bistu.edu.cn

1 Beijing Key Laboratory of High Dynamic Navigation Technology, Beijing Information Science & Technology University, Beijing 100192, China

2 Key Laboratory of Modern Measurement and Control Technology, Ministry of Education, Beijing 100192, China

3 School of Automation, Beijing Information Science & Technology University, Beijing 100192, China

4 School of Information and Communication Engineering, Beijing Information Science & Technology University, Beijing 100101, China

with an excessive number of dynamic objects. Furthermore, when confronted with more intricate environments, the computational resources required for maintaining a substantial number of keyframes become a limiting factor.

The DynaSLAM system, as proposed by Bescos et al. [3], incorporates a segmentation network utilizing a pixel-level Mask R-CNN model [9]. This model is proficient at segmenting objects within each frame at the pixel level. Additionally, the system introduces a multi-view geometry approach for the secondary identification of potential non-stationary objects in the image, resulting in a substantial enhancement in accuracy. Nevertheless, it is worth noting that the requirement for image data segmentation in every frame poses a notable challenge to the real-time performance of the system.

Zhang et al. [10] introduced the WF-SLAM system, which primarily integrates Mask-RCNN networks with geometric methods to tightly couple the extraction of feature information for every object in the environment and subsequently allocate weights. This approach is advantageous for precise segmentation of dynamic objects, enhancing accuracy. While the system employs object detection to assist in detection and avoid full-scene segmentation, it still inevitably involves segmenting each frame's image data, and there is the potential for redundant computations when detection boxes overlap.

JCV Soares et al. [11] employ object detection to identify individuals within the scene and subsequently perform a complete removal of feature points located within the detected bounding boxes. This approach of excessive feature point removal may lead to the inadvertent deletion of stationary objects contained within the bounding boxes, thereby potentially adversely affecting the accuracy of pose estimation and mapping.

In summary, while contemporary SLAM algorithms designed for dynamic environments can detect and eliminate select dynamic feature points, they suffer from issues such as high computational requirements, Excessive culling of feature points, and inadequate real-time performance. Furthermore, researchers have not yet delved deeply into the study of indoor crowds. This has resulted in recurrent challenges for intelligent robots when navigating through human traffic in indoor settings. For instance, issues may arise during the transportation of goods, such as the failure to recognize people, leading to potential collisions. Therefore, research in the domain of indoor crowd environments has emerged as a critical concern.

Therefore, this paper proposes a new SLAM system called CP-SLAM (Crowdplus-SLAM), an ORB-SLAM2 [1] system specifically improved for Dynamic crowd environments. It can achieve efficient and accurate SLAM localization in crowd scenes without pre-training. We adopt a new technique to solve traditional algorithms' dynamic feature point removal and crowd detection problems. The main contributions of this paper can be summarized as follows:

1. In the improved framework of ORB-SLAM2, a novel target detection thread is introduced. The target detection network has been enhanced to increase its detection coverage in complex crowd environments.
2. We propose a novel feature-filtering algorithm that integrates information from detection boxes and geometric depth to preserve useful static features extracted from the detection framework while rejecting dynamic information. This approach allows for better preservation of static features and improves the robustness of the feature filtering process.
3. We evaluated the performance of our proposed algorithm, CP-SLAM, on two dynamic RGB-D datasets: TUM and Bonn RGB-D. The positional and trajectory errors of CP-SLAM were compared with those of ORB-SLAM2 and other state-of-the-art algorithms.

The rest of this paper is organized as follows. Section II describes the system framework. In Section III, the proposed feature filtering algorithm is explained. Section IV shows the experimental comparison as well as the analysis. In Section V, the conclusions are given.

2 System Framework

ORB-SLAM2 is a visual SLAM system with multiple threads, and multiple threads can be parallelized at the same time. This system shows high accuracy and robustness in a static environment, and the drawback is that it does not perform well in dynamic scenes. Therefore, in this paper, the CP-SLAM framework is improved based on ORB-SLAM2. The general framework is shown in Fig. 1, with the Dynamic Target Detection thread on the far left, the Tracking thread on the top, the Local Mapping thread on the far right, and the Loop Detection thread on the bottom.Our work focuses on enhancing the processing capability of the tracking and target detection threads in a dense crowd dynamic environment to improve the overall system performance. In the feature extraction stage, we extract feature point distribution information of crowds and other objects in the image. We then integrate a target detection thread to receive people's detection frame information. In the subsequent processing stage, we pass the detection frame information to the tracking thread, which combines it with the feature point distribution information. We employ a feature filtering method to remove the dynamic "human" feature information. The resulting feature information is then passed to the local map tracking and new keyframe generation.

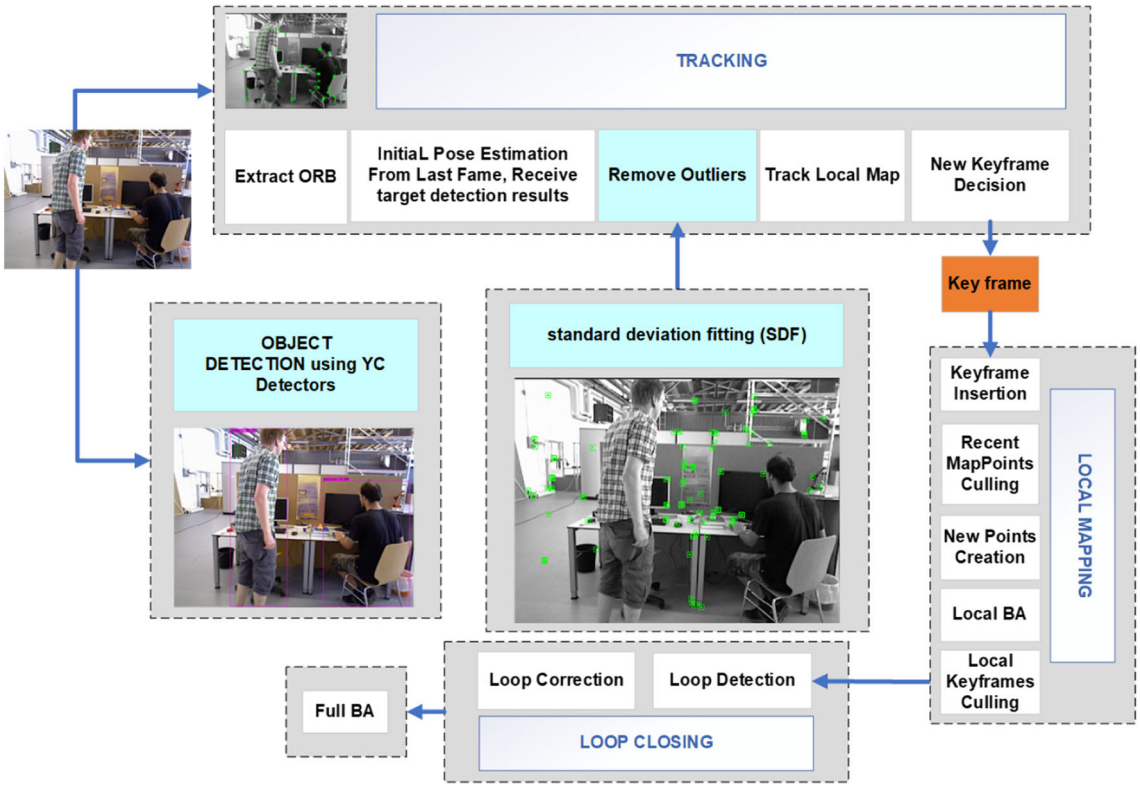


Fig. 1 System framework (Improvements on blue background)

In the LocalMapping thread, the local map will acquire keyframes and filter the map points for rejection. The local beam levelling (BA) is then used to adjust the poses further and improve the map points. Finally, the keyframes are re-filtered

The loopback detection thread contains two main parts: loopback detection and loopback correction, which mainly detect and correct key frame information. Finally, the position and attitude are optimized through global BA.

3 Feature Filtering Algorithm

This section describes the target detection network we used and the proposed feature filtering algorithm.

3.1 Target Detection Networks

When confronted with congested environments, the utilization of an instance segmentation model for scene analysis can substantially escalate the computational load, consequently impeding real-time operational efficiency. In light of this challenge, we propose the adoption of target detection as a solution. Specifically, we employ the lightweight YOLOv4-tiny [12] detection network, which is devoid of dependence on deep layers and endows the system with rapid inference capabilities.

In response to the restricted expressive capacity of LeakyReLU in accommodating intricate data distributions and capturing non-linear associations, this investigation introduces the Swish activation function as a replacement. The mathematical expression of the Swish function is Eq. 1 and its derivative is Eq. 2. Swish manifests nonlinearity over non-negative input values, thereby augmenting the model's capability to adapt to complex data distributions. Comparative diagrams of activation functions are delineated in the subsequent Fig. 2. Furthermore, to enhance the detection coverage in dense crowds, the dual YOLO output layers were replaced with three YOLO output layers.

$$f(x) = x \cdot Sigmoid(x) = \frac{x}{1 + e^{-x}} \tag{1}$$

$$f'(x) = f(x) + \frac{e^{-x}}{1 + e^{-x}} \cdot (1 - f(x)) \tag{2}$$

To optimize the network's computational speed and efficacy, the YOLOv4-tiny detection network is trained employing the CrowdHuman dataset [13], culminating in the development of our proposed network architecture, denoted as YC (YOLOv4-tiny for CrowdHuman). Our improved method achieves fast and accurate crowd detection, out-

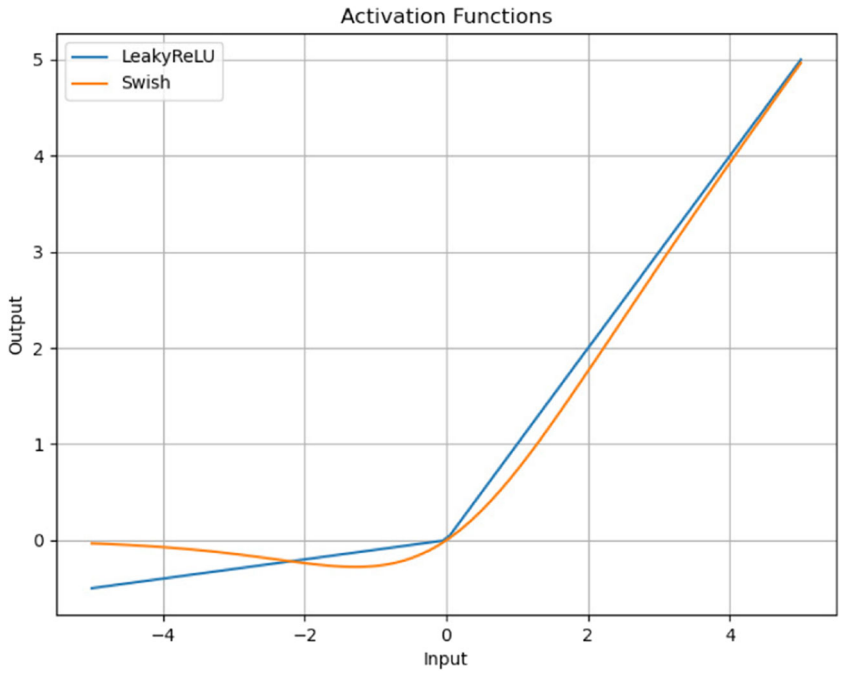


Fig. 2 Activation function graph

performing traditional methods while maintaining real-time performance in crowded environments. The network structure diagram of YC is shown in Fig. 3.

Table 1 provides a comparison between the YOLOv4-tiny network and the enhanced YC network. From the data, it is evident that our network exhibits a substantial improvement in average accuracy, with a remarkable 45.7% increase. Simultaneously, the recall rate has also seen a 45.7% enhancement. This suggests an overall superior detection performance, capable of identifying a greater number of true positive samples and thereby enhancing detection coverage. However, it's worth noting that the introduction of some false positives within the positive samples has led to a slight decline in precision.

The MOT dataset includes scenes with large crowds gathering, such as train stations, where the number of people occupies more than 2/3 of the scene, which is in line with the scenario of our research. We select a frame from the video of the MOT20-2 sequence in the MOT dataset [14] for detection. The test results are shown in Fig. 4. The results indicate

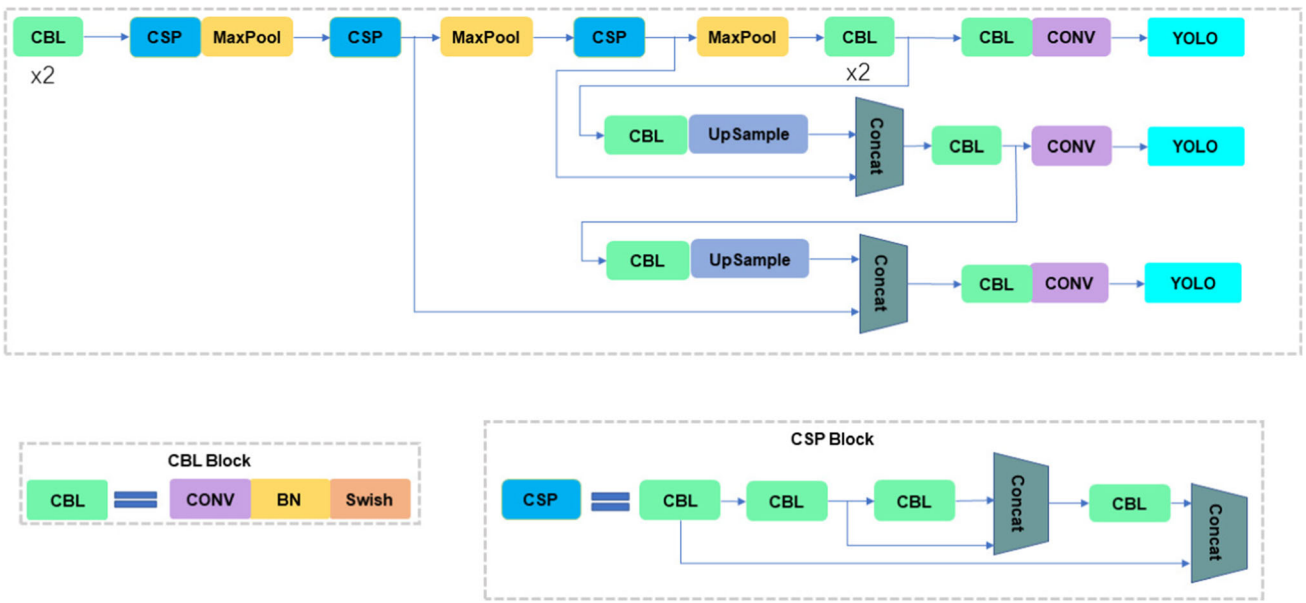


Fig. 3 YC network structure map

Table 1 Network evaluation indicators

	mAP@0.50	Precision	Recall
yolov4-tiny	36.02%	0.69	0.35
YC	50.97%	0.65	0.51

that the enhanced network achieved a detection coverage rate of over 95% and an accuracy exceeding 96%.

3.2 Dynamic Point Rejection Algorithm

The traditional approach based on target detection is to inspect the image to get the anchor box of each class, then input the information of the box to the Tracking thread, and then reject all the feature point information in the dynamic object box without filtering. However, existing methods for processing dynamic feature points suffer from significant drawbacks, as shown in Fig. 5. When a static object is located within a dynamic object box, or when a part of the object classified as static is in a dynamic object box, the feature points that should be left on the static object are directly rejected. The result is an increase in the error of the positional estimation, which reduces the system's positioning accuracy and map-building integrity. To solve this problem, We propose a novel approach that combines depth values of feature points to segregate and eliminate non-static feature points, which involves a feature point filtering algorithm founded on standard deviation fitting (SDF).

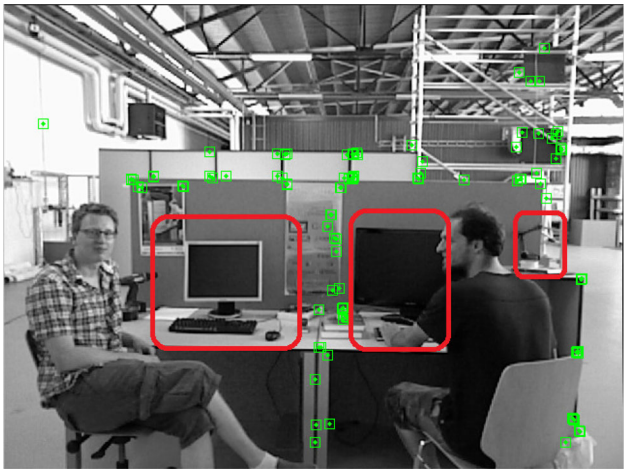


Fig. 5 Feature point misdeletion map.(Error deletes feature points in the red box that are not on the person)

In-depth images, there is a significant disparity in depth between the human body and other objects. Furthermore, humans often occupy a substantial portion of the image and their depth values are well-distributed. Hence, employing pixel-wise depth segmentation of humans is a viable approach. To remove outliers, we adopt the Maximum Likelihood Estimation Sample Consensus (MLESAC) [15] algorithm, which differs from the traditional fitting methods

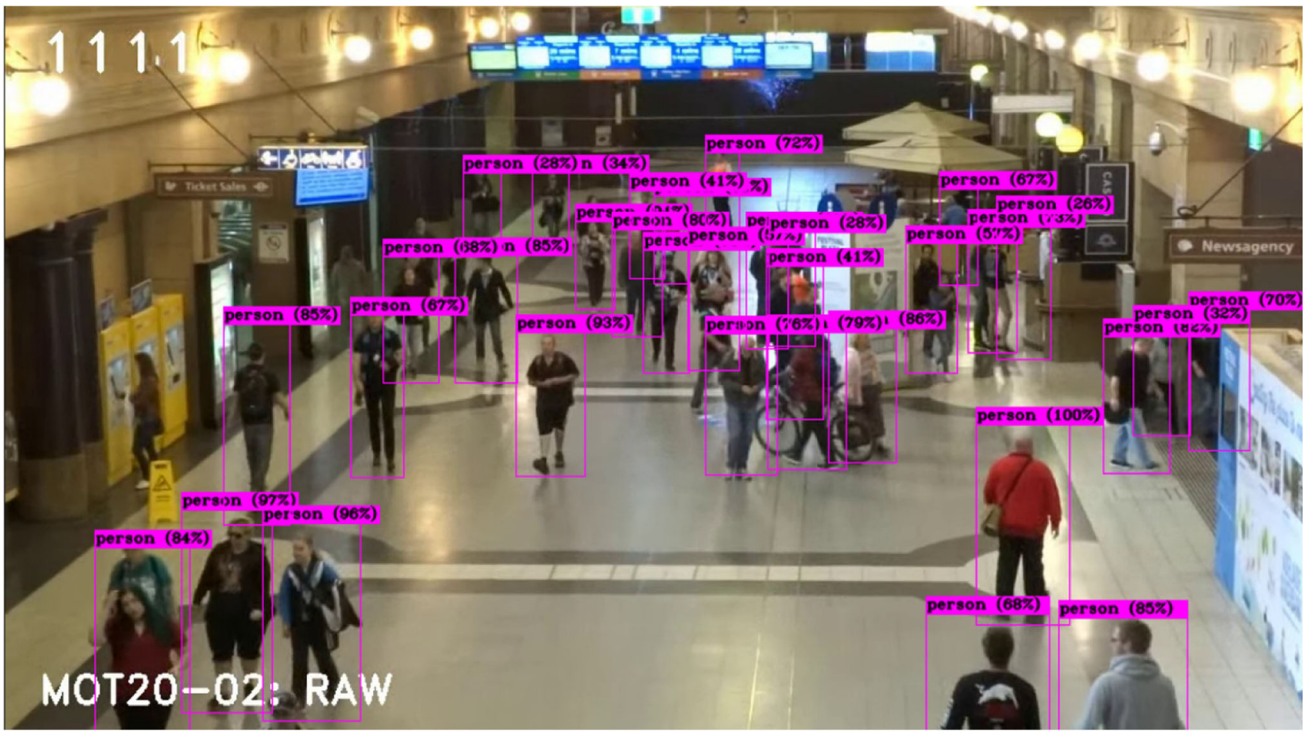


Fig. 4 YC test map

in that we do not use a linear model, but instead choose the standard deviation of pixel depth as the criterion for differentiation, and then calculate the likelihood value to distinguish between inliers and outliers. This improves the robustness and flexibility of the algorithm.

The SDF is mainly divided into two steps, firstly, this paper has to traverse each detection frame to determine whether the feature point is inside the frame, and if the feature point is inside the target frame, this paper stores it in a dynamic point set. After that, this paper randomly selects two points in the dynamic set, calculates the standard deviation between their depth values as a mathematical calculation model, and then iterates through all the feature points in the dynamic point set, and stores the points within the range as "good points" in another set, so that this paper gets a set of feature points that meet the requirements. The points within this set are the feature points within the contour of the non-static object that we do not need, and all remaining feature points are the static points used for subsequent threads. The algorithm is as follows Algorithm 1.

Algorithm 1 Standard deviation fitting.

Input: $npoints$: Feature points,$nboxes$:Number of boxes, $nIter = 0$:Number of iterations,th:thresholds
Output: Static point set:$S - S1$

```
for i = 0; i < npoints; i++ do
    if i in boxes then
        i add to S
    end if
end for
while nIter < k : do
    Randomly select d1 and d2 from S;
    Calculate the standard deviation of the d1, d2 depth values SD0;
    for j in S − (d1 + d2): do
        Calculate the standard deviation between their depth values SD;
        Calculate likelihood ;
        if likelihood < th : then
            nInliers++
            j add to S1
        end if
    end for
    nIter++
end while
```

The formulas in the algorithm contain the following:

$$\mu = \frac{\sum_{i=1}^{n}(x_i)}{n} \tag{3}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n}} \tag{4}$$

$$L = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{5}$$

where σ denotes the standard deviation and μ denotes the mean value,L is likelihood refers to the value of the probability density function.

Considering that setting a large value for K could significantly increase the computational burden on the system, leading to delays in processing, we have chosen to set $K = 20$. The selection of the appropriate threshold, th, is crucial for robustly identifying inliers and fitting a suitable model, while effectively excluding noise and outliers. In this paper, We have devised a novel method for determining the threshold value, as represented in Formula 6. This approach enables flexible adjustment of the threshold value based on the changing relative density of inliers, thereby mitigating errors associated with a fixed threshold value.

$$th = 0.02 * (1.0 + nInliers/S.size()) \tag{6}$$

Here, $nInliers$ refers to the number of inliers, and $S.size()$ indicates the number of feature points within the detection box.

4 Experiments and Analyses

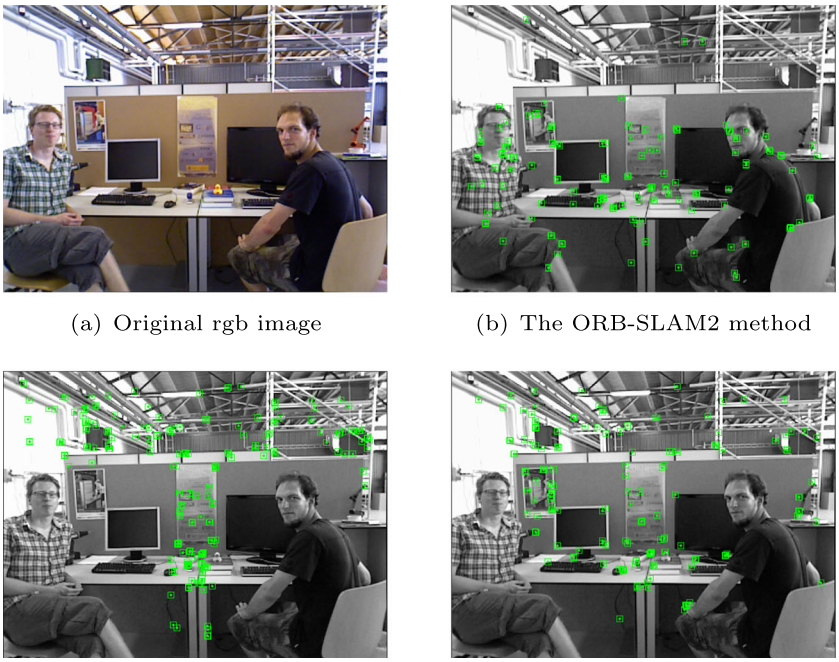
In this section, we first compare CP-SLAM with ORB-SLAM2 and then compare it with other superior dynamic SLAM algorithms to highlight the system's performance. We first select the RGB-D TUM Dataset [16] for experimental comparison and analysis and then further validate it on the Bonn RGB-D Dynamic Dataset [17]. Absolute trajectory error (ATE) is chosen to compare the deviation degree of trajectory of each system, and relative pose error (RPE) is used as the evaluation index of pose estimation, and root mean square error (RMSE) and mean value (Mean) under the two errors are used as the metric.

The experiments were conducted on a laptop with Ubuntu 20.04 OS, NVIDIA GeForce GTX 1650, and Intel Core i5-10200H CPU.

4.1 TUM Dataset Validation

The TUM RGB-D dataset has become the primary experimental dataset in the field of SLAM, which contains depth images, RGB images, accurate ground data, and actual dynamic data. The dataset includes a variety of scenes, including most real-life human sitting and walking postures, as well as objects such as tables and monitors. In this paper, several of them and representative ones containing walking and other highly dynamic sequences are selected for experimental comparison.

Figure 6 shows the comparison of feature detection under fr3_walking_xyz high dynamic sequence. From the chart can be seen that this paper's method (Fig. 6d) compared to ORB-



(a) Original rgb image (b) The ORB-SLAM2 method

(c) Only target detection results (d) Our

Fig. 6 Comparison of feature detection. (In Figure d, not only are the feature points on individuals removed, but also those on objects such as screens and mice are retained)

SLAM2 (Fig. 6b), this paper on the scene of people and objects in the dynamic and static object classification, the human as dynamic, the extracted feature points almost no longer on the person, compared to completely remove the box feature points in Fig. 6c, this paper also retains the feature points on static objects such as monitor, mouse, and keyboard, which will not cause excessive waste of feature points.

In Fig. 7, ORB-SLAM2 in the left column, DynaSLAM in the second column, and CP-SLAM in the third column are selected for comparison, and the ATE and RPE plots of the fr3_walking_static sequence (blue plots) are used to

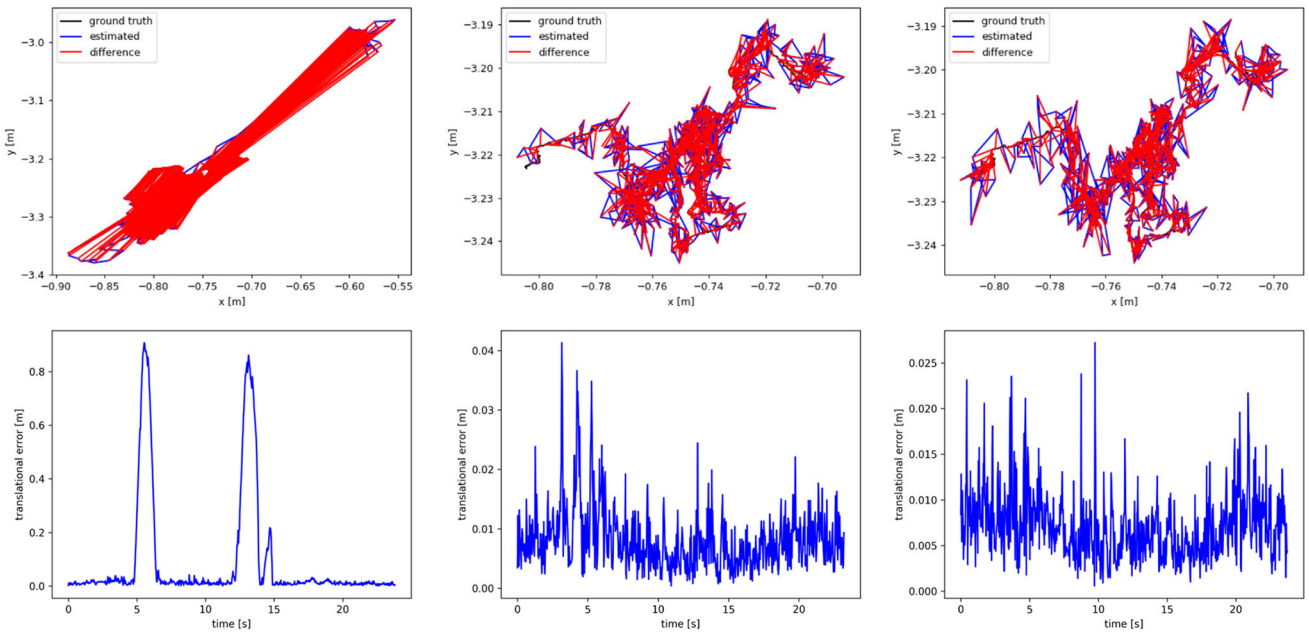
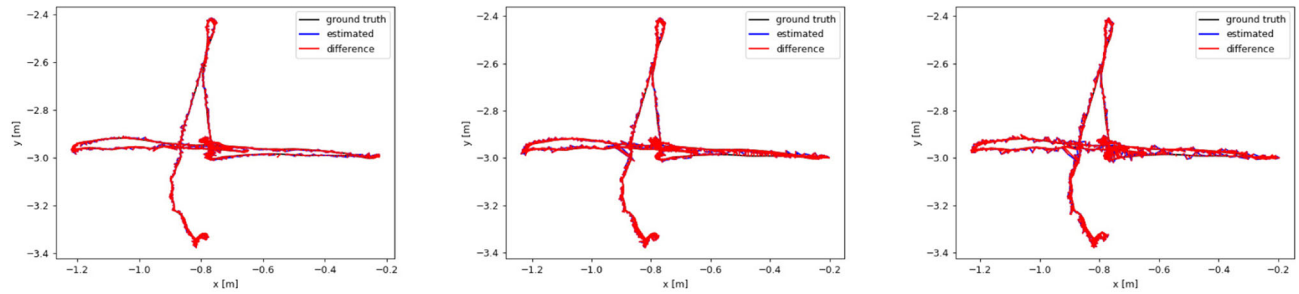
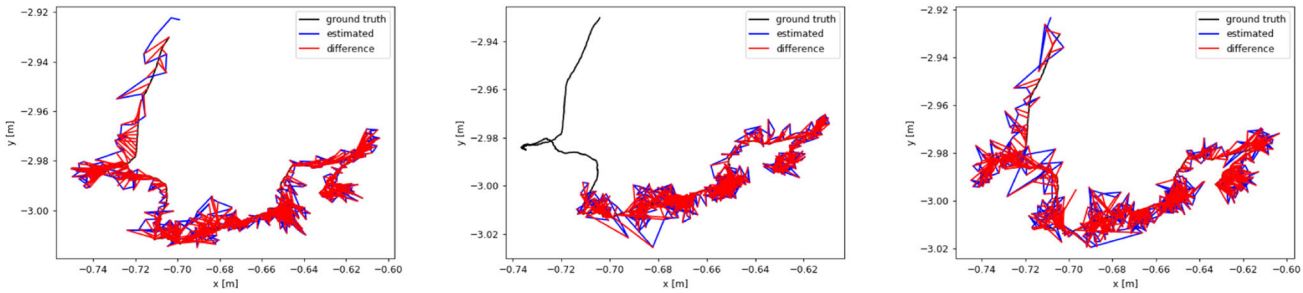


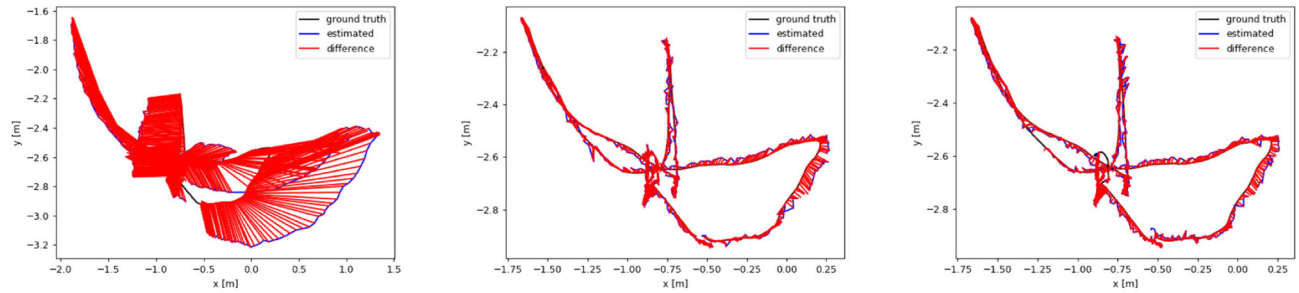
Fig. 7 Performance comparison on fr3_walking_static sequence. (In the comparative chart with the blue curve, the errors of our algorithm are consistently below 0.025, which is significantly smaller than the errors observed in the other two systems)



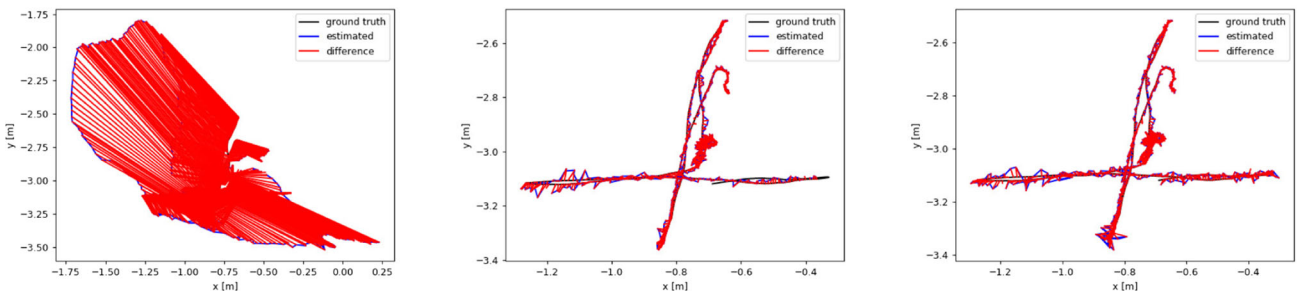
(a) fr3_sitting_xyz sequence



(b) fr3_sitting_static sequence



(c) fr3_walking_halfsphere sequence



(d) fr3_walking_xyz sequence

Fig. 8 Comparison of absolute trajectory error of ORB-SLAM2 (left) with DynaSLAM (middle) and Our (right)

Table 2 Comparison of absolute trajectory errors between the CP-SLAM and ORB-SLAM2 algorithms of this paper (ATE unit: m)

Sequence	ORB-SLAM2		CP-SLAM		Improvements/%	
	RMSE	Mean	RMSE	Mean	RMSE	Mean
fr3_sitting_static	0.008	0.007	0.007	0.006	**12.50**	**14.29**
fr3_sitting_xyz	0.009	0.008	0.014	0.013	−35.71	−62.50
fr3_sitting_rpy	0.018	0.014	0.013	0.012	**27.78**	**14.29**
fr3_sitting_half	0.021	0.017	0.017	0.015	**23.81**	**11.76**
fr3_walking_static	0.377	0.341	0.006	0.005	**98.41**	**98.53**
fr3_walking_xyz	0.739	0.624	0.018	0.015	**95.56**	**97.60**
fr3_walking_rpy	0.469	0.423	0.030	0.024	**93.60**	**94.33**
fr3_walking_half	0.442	0.358	0.028	0.025	**93.67**	**93.02**

Bolded text indicates smaller error data or a boost greater than zero

Table 3 Comparison of absolute trajectory errors between the CP-SLAM and ORB-SLAM2 algorithms of this paper (ATE unit: m)

Sequence	Liu et al.	DS-SLAM	DynaSLAM	Crowd-SLAM	CP-SLAM
	RMSE/Mean	RMSE/Mean	RMSE/Mean	RMSE/Mean	RMSE/Mean
fr3_sitting_static	**0.006/0.005**	**0.006**/0.006	**0.006/0.005**	0.008/0.007	0.007/0.006
fr3_sitting_xyz	−/−	−/−	**0.014/0.013**	0.018/0.017	**0.014/0.013**
fr3_sitting_rpy	−/−	−/−	0.046/0.026	0.015/0.013	**0.013/0.012**
fr3_sitting_half	−/−	−/−	0.022/0.018	0.020/0.017	**0.017/0.015**
fr3_walking_static	0.010/0.007	0.008/0.007	0.007/0.006	0.007/0.007	**0.006/0.005**
fr3_walking_xyz	0.016/0.014	0.024/0.019	**0.015/0.013**	0.020/0.017	0.018/0.015
fr3_walking_rpy	0.042/0.030	0.444/0.377	−/−	0.044/0.031	**0.030/0.024**
fr3_walking_half	0.031/0.026	0.030/0.026	0.030/0.026	**0.026/0.022**	0.028/0.025

Bolded text indicates smaller error data or a boost greater than zero

Table 4 Comparison results and improvement of absolute trajectory error (ATE [m])

Sequence	RDS[1]	Improve[2]	SOLO-SLAM		Improve		CP-SLAM	
	RMSE	RMSE	RMSE	Mean	RMSE	Mean	RMSE	Mean
fr3_W[3]_static	0.0206	**70.87%**	0.0104	0.0093	**42.31%**	**44.09%**	0.0060	0.0052
fr3_W_xyz	0.0571	**67.78%**	0.0187	0.0167	**1.60%**	**5.39%**	0.0184	0.0158
fr3_W_rpy	0.1604	**81.30%**	0.1194	0.0894	**74.87%**	**72.37%**	0.0300	0.0247
fr3_W_half	0.0807	**64.44%**	0.0276	0.0240	−3.98%	−3.75%	0.0287	0.0249

[1] RDS is RDS-SLAM
[2] Improve is Improvements
[3] W is walking
Bolded text indicates smaller error data or a boost greater than zero

Table 5 Root mean square error and mean value of translational drift in RPE

Sequence	ORB-LAM2	DS-SLAM	DynaSLAM	CP-SLAM
	RMSE/Mean	RMSE/Mean	RMSE/Mean	RMSE/Mean
fr3_sitting_static	0.012/0.011	**0.008/0.007**	0.009/0.008	0.010/0.009
fr3_sitting_xyz	**0.014/0.012**	−/−	0.021/0.019	0.021/0.019
fr3_sitting_rpy	0.027/0.023	−/−	0.067/0.042	**0.023/0.020**
fr3_sitting_half	0.031/0.025	−/−	0.030/0.027	**0.010/0.009**
fr3_walking_static	0.540/0.366	0.010/0.009	**0.009**/0.008	**0.009/0.006**
fr3_walking_xyz	1.116/0.908	0.033/0.024	**0.022/0.019**	0.026/0.023
fr3_walking_rpy	0.687/0.588	0.150/0.094	−/−	**0.059/0.049**
fr3_walking_half	0.649/0.505	**0.030/0.026**	0.043/0.038	0.041/0.036

Bolded text indicates smaller error data or a boost greater than zero

Table 6 Root mean square error and mean value of rotational drift in RPE

Sequence	ORB-LAM2	DS-SLAM	DynaSLAM	CP-SLAM
	RMSE/Mean	RMSE/Mean	RMSE/Mean	RMSE/Mean
fr3_sitting_static	0.348/0.314	**0.273/0.245**	0.327/0.293	0.317/0.283
fr3_sitting_xyz	**0.585/ 0.497**	–/–	0.624/0.543	0.592/0.514
fr3_sitting_rpy	0.847/0.747	–/–	1.019/0.862	**0.618/0.553**
fr3_sitting_half	0.769/0.697	–/–	0.810/0.718	**0.317/0.283**
fr3_walking_static	9.868/6.721	0.269/0.242	0.327/0.293	**0.264/0.238**
fr3_walking_xyz	7.302/5.652	0.826/0.584	**0.614/0.486**	0.668/0.530
fr3_walking_rpy	11.778/9.711	3.004/1.919	–/–	**1.393/1.185**
fr3_walking_half	15.483/12.684	**0.814/0.703**	0.972/0.868	0.959/0.855

Bolded text indicates smaller error data or a boost greater than zero

visualize the performance of each system. where the true value is in black and the estimated value is in blue, and the deviation of the true trajectory from the estimated trajectory is in red. Therefore, the length of the red curve represents the greater deviation from the real camera trajectory. The deviation degree of the camera motion trajectory of our system in this figure is much smaller than that of the ORB-SLAM2 system and close to that of the DynaSLAM system. Therefore, it shows that the accuracy of our system is better than ORB-SLAM2 and not lower than the DynaSLAM system. The horizontal coordinate in the RPE plot represents time in s; the vertical coordinate represents the value of positional error in m, and the higher the curve, the larger the error. the error range of ORB-SLAM2 in the RPE plot is 0-0.8; the error range of DynaSLAM positional error is 0-0.04, concentrated between 0.005-0.02; CP-SLAM's positional error range is 0-0.025, with error values concentrated between 0.0025-0.015. Therefore, the RPE curves can be seen in this paper that the system performance is a little better than the first two.

Figure 8 compares the absolute trajectory error of our proposed CP-SLAM (right) with ORB-SLAM2 (left) and DynaSLAM (middle) on the selected sequence. The study compares the performance of xyz and static sequences in fr3_sitting, along with halfsphere and xyz sequences in fr3_walking. The shorter red curve corresponds to a smaller error. From the comparison chart, it is easy to see that the CP-SLAM after filtering the redundant features is smaller in error than the other two, and the pose estimation will be more accurate.

CP-SLAM and ORB-SLAM2 were selected for comparative experimental analysis in Table 2. By comparing the RMSE and Mean values under ATE with multiple sequences on the dataset. The data are marked boldly to indicate better accuracy, which is the case for all of the following. We selected the fr3_walking_rpy sequence as highly dynamic because its scene is continuously and substantially moving with people, etc. In this high active sequence comparison, we found that the RMSE of CP-SLAM improved by 93.60%, and the Mean value improved by 94.33%. The average improvement of RMSE in the walking high dynamic sequence is 95.93%. This suggests that CP-SLAM can address the interference caused by the presence of dynamic objects in the scene.

In Table 3, CP-SLAM is selected for comparative analysis with Liu, Detect-SLAM, DS-SLAM, DynaSLAM, and Crowd-SLAM [11]. Where "-" indicates no analysis result for this dataset in this literature. The comparison results of ATE data from the selected sequences show that the error of our CP-SLAM is better than the other algorithms in most environments, indicating that our algorithm has higher accuracy under these sequences. Although there are still small parts that are still inferior, the difference between them is tiny.

Table 4 shows the absolute trajectory error and the improvement in the high dynamic series between CP-SLAM and the two recent years of the excellent performance of RDS-SLAM [18] and SOLO-SLAM [19] systems. The experimental data show that the REME values obtained by our system improve over RDS-SLAM for all four series selected from the TUM dataset, by at least 64.44% and up to 81.30%. In comparison with the latest SOLO-SLAM, our MEAN and RMSE are higher than it except for the fr3_walking_half sequence. The experiments show that our proposed feature point rejection algorithm can achieve the desired effect of separating dynamic and static objects so that the system's performance in the dynamic environment is no less than that of similar systems and thus meet real-life applications.

The main reason why CP-SLAM can perform well in high dynamic environments is its ability to accurately acquire target detection frames and the algorithm's ability to accurately and not excessively reject dynamic features in the target frame, making its acquisition of keyframes more accurate.

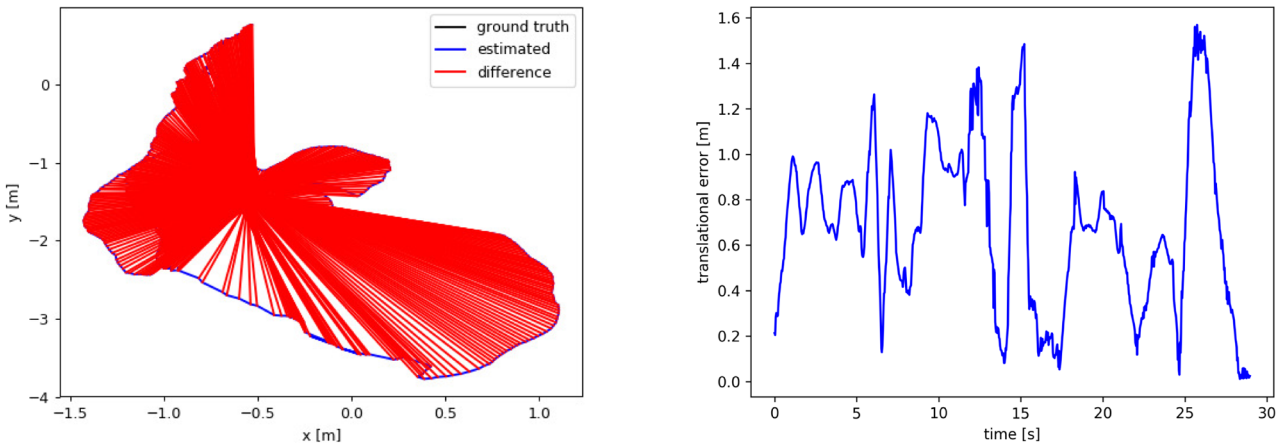
The relative positional errors of the algorithm in this paper are tested below, comparing this paper with ORB-SLAM2, DS-SLAM, and DynaSLAM systems, respectively. Table 5 shows the comparison results of translational and rotational drift on eight sequences selected from the TUM

dataset are shown in Table 6. The comparison shows CP-SALM is slightly better under high dynamic sequences such as fr3_walking_hal and does not lag too much under other lower dynamic sequences.

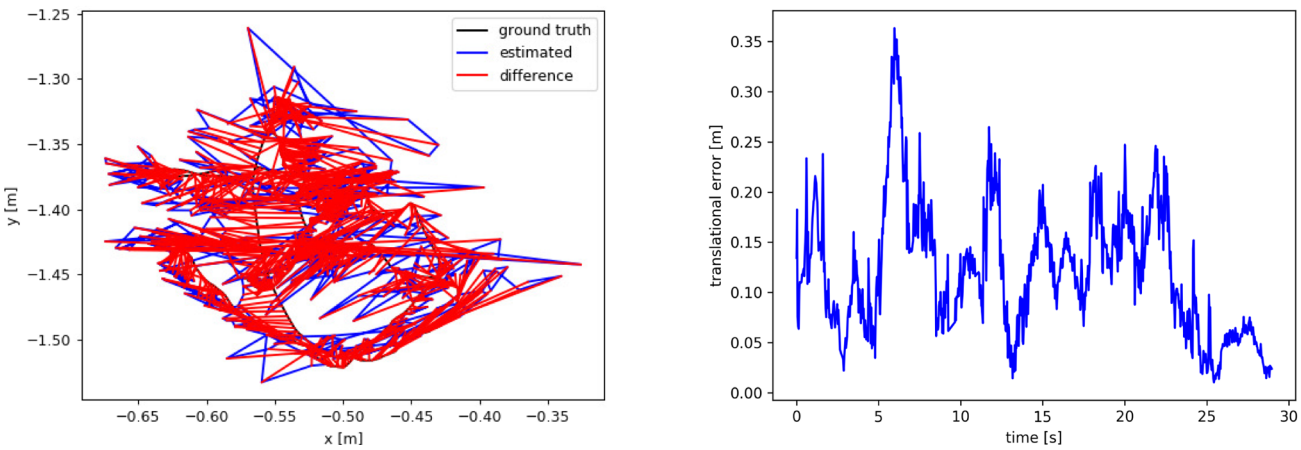
4.2 Bonn RGB-D Dynamic DataSet

Although the TUM dataset is widely used as a benchmark for VSLAM systems, it remains unconvincing for applications in crowded environments such as supermarkets, stations, etc., because the number of people in their dynamic scenes is too small. Therefore, the Bonn RGB-D dynamic dataset is selected for further testing, which contains a more significant number of people and people walking, which is closer to the environment of dense crowds.

In Fig. 9, this paper compares the system performance with ORB-SLAM2 and this paper's algorithm CP-SLAM for the selected crowd2 sequences to further demonstrate the superior system performance in a dense environment. From the ATE curve in the left half of the figure, we can see that the red curve length of CP-SLAM is smaller than that of the ORB-SLAM2 system, indicating that the CP-SLAM system has less error in this population environment. The blue curve in the right half of the figure shows that the RPE values of CP-SLAM are between 0.1 and 0.36, but the error value of the ORB-SLAM2 system is up to 1.6. The above comparison

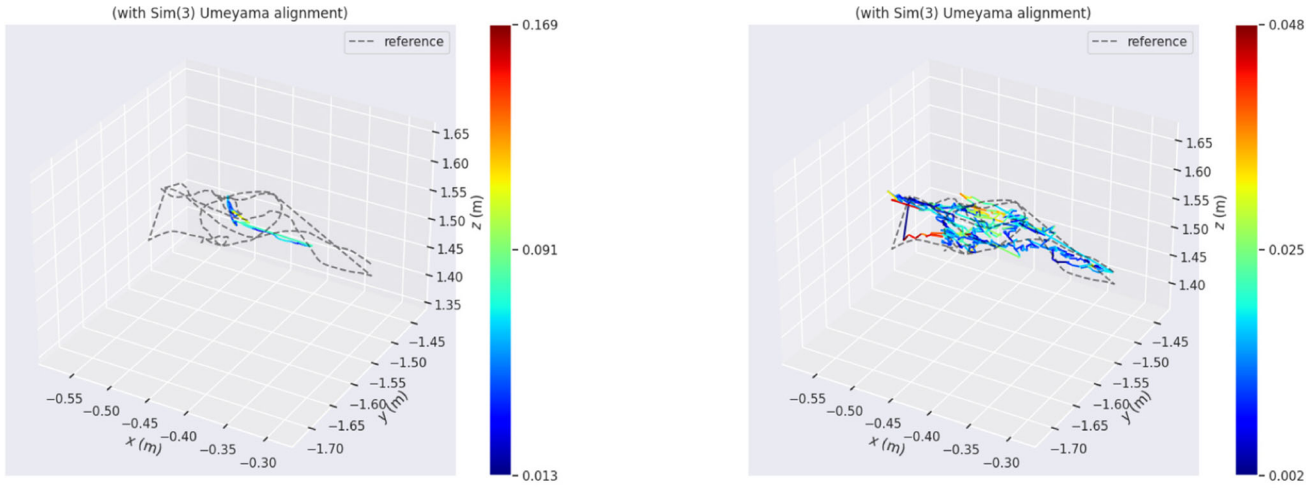


(a) ORB-SLAM2

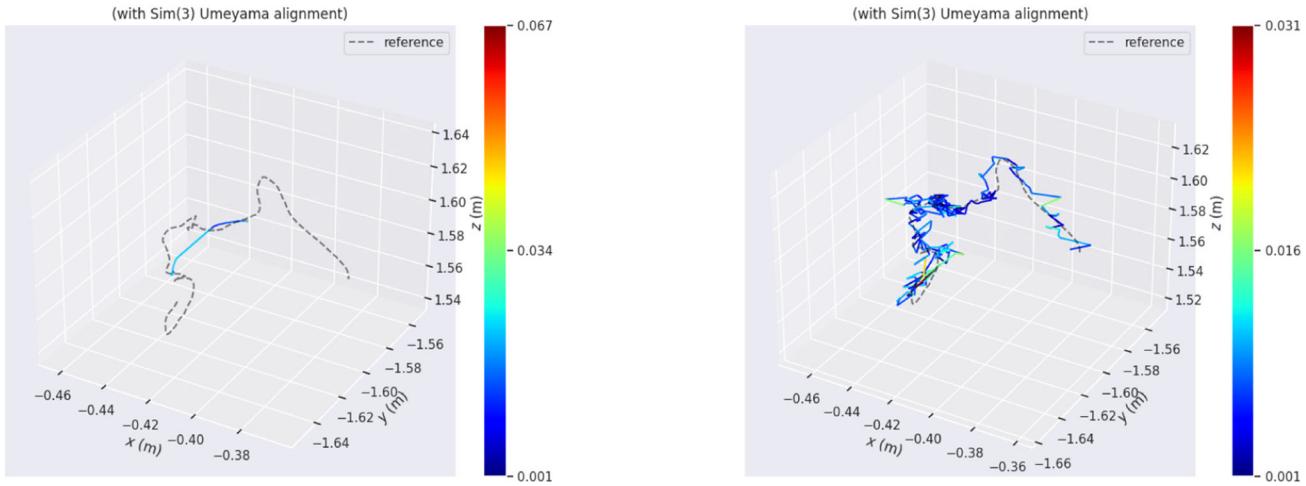


(b) Our

Fig. 9 Comparison of crowd2 sequence performance. (In the comparative chart, the blue curve representing our algorithm consistently maintains errors below 0.35, which are substantially lower than the errors exhibited by the other system)



(a) crowd1 sequence



(b) synchronous2 sequence

Fig. 10 Comparison of the pose trajectory error.(From the 3D trajectory comparison, it is evident that the overlap of our algorithm significantly surpasses that of the ORB-SLAM2 system)

can show that our system has higher accuracy in a dynamic crowd environment.

This paper evaluates the pose trajectory error of our proposed CP-SLAM algorithm (right) and ORB-SLAM2 (left) on the selected crowd1, crowd3, and synchronous2 sequences in Fig. 10. As the comparison graph shows, our system's trajectory more closely approximates the camera's true trajectory, resulting in significantly lower pose error. This demonstrates that our algorithm effectively reduces the impact of dynamic objects on the scene and improves system accuracy.

We selected the Bonn RGB-D dynamic dataset crowd1-3, synchronous1-2 sequences for experimental comparison. These five sequences have the characteristics of a large number of people and high dynamics. From the comparison of RMSE values under ATEA in Table 7, we can find that the values of CP-SLAM are much smaller than ORB-

Table 7 RMSE for absolute trajectory error

Sequence	ORB-SLAM2	CP-SLAM	Improvements
crowd1	0.914	**0.026**	97.16/%
crowd2	1.453	**0.037**	97.45/%
crowd3	1.140	**0.033**	97.11/%
Synchronous1	1.098	**0.009**	99.18/%
Synchronous2	1.484	**0.008**	99.46/%

Bolded text indicates smaller error data or a boost greater than zero

Table 8 Comparison of root mean square errors of RPE

Sequence	ORB-SLAM2		CP-SLAM		Improvements/%	
	LD[1]	Rd[2]	LD	Rd	LD	Rd
Crowd1	1.324	39.463	0.317	21.040	**76.06**	**46.68**
Crowd2	2.105	85.735	0.218	25.315	**89.64**	**70.47**
Crowd3	1.695	42.485	0.232	23.916	**86.31**	**43.71**
Synchronous1	1.551	27.880	0.028	1.272	**98.19**	**95.44**
Synchronous2	2.139	38.688	0.008	9.289	**99.63**	**75.99**

[1] LD is Levelling Drift
[2] Rd is Rotational drift
Bolded text indicates smaller error data or a boost greater than zero

SLAM2. The error is reduced by at least 97.11% compared to ORB-SLAM2. The translational drift error and rotational drift error of the two systems are compared separately in Table 8. Both error values are reduced under CP-SLAM, where the translational drift error is reduced by at least 76.06%, and the rotational drift error is reduced by 43.71% to 95.44%. In summary, the experimental comparison between the two tables shows that our system can reduce the effect of dynamic population on the positional estimation, thus improving the system's accuracy.

4.3 Real-Time Operation

We establish a real-time transmission channel that enables the rapid transmission of detection frame information during the detection process while using the rejection algorithm to quickly estimate the pose of the scene and build the map. This approach eliminates the need for pre-training the scene, making it highly efficient and effective for real-time dynamic SLAM applications.In addition, we compared the running times of ORB-SLAM2, CP-SLAM, and Dyna-SLAM on the fr3_walking_xyz sequence, as shown in Table 9. As you can see, CP-SLAM have the same good real-time performance as ORB-SLAM2.

5 Conclusions

In this study, we present a novel dynamic SLAM system built upon the ORB-SLAM2 algorithm, aimed at enhancing the practicality of SLAM in crowded environments. This new system incorporates a target detection thread to identify dynamic individuals within the scene. We introduce a feature point filtering algorithm based on the standard deviation fitting (SDF) to eliminate feature points located on individuals within the detection boxes, thus preventing the issue of indiscriminate removal of static points within the detection boxes. Experimental validation on two datasets demonstrates that our system substantially reduces the ATE and RPE errors by over 90% when compared to ORB-SLAM2 in highly dynamic crowd environments. This indicates its high precision and suitability for indoor crowd scenarios, such as shopping malls and train stations.

Nevertheless, the system does have certain limitations, such as the less pronounced improvements in low-dynamic environments. In the future, our research will continue to focus on gradually enhancing its applicability in various settings.

Table 9 Running time

System	ORB-SLAM2	DynaSLAM	CP-SLAM
Running time(s)	38'24	412'26	39'24

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant Nos. 61703040 and 61603047) and the Teacher Recruitment and Support Plan of Beijing Information Science & Technology University (Grant No. 5029011103). The authors thank Juan Dai for her technical assistance with the experiments and Zhong Su and Cui Zhu for their insightful suggestions throughout the study. The authors also express their gratitude to the anonymous reviewers for their valuable comments and suggestions that helped improve the quality of this paper.

Author Contributions Jianfeng Li served as the first author and completed the entire creative process. Dai Juan acted as the corresponding author, providing guidance and revising the manuscript. Zhong Su and Cui Zhu contributed to the revision of the manuscript as assisting authors.

Code or data availability The Crowdhuman dataset [13], MOT dataset [14], TUM dataset [16], and Bonn RGB-D Dynamic DataSet [17] were used in our work.

Declarations

Conflicts of interest/Competing interests Not applicable

Ethics approval Not applicable

Consent to participate Not applicable

Consent for publication Not applicable

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

1. Mur-Artal, R., Tardós, J.D.: Orb-slam2: an open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Trans. Rob. **33**(5), 1255–1262 (2017). https://doi.org/10.1109/TRO.2017.2705103
2. Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M., Tardós, J.D.: Orb-slam3: an accurate open-source library for visual, visual-inertial, and multimap slam. IEEE Trans. Rob. **37**(6), 1874–1890 (2021)
3. Bescos, B., Fácil, J.M., Civera, J., Neira, J.: Dynaslam: tracking, mapping, and inpainting in dynamic scenes. IEEE Robot. Autom. Lett. **3**(4), 4076–4083 (2018). https://doi.org/10.1109/LRA.2018.2860039
4. Li, S., Lee, D.: Rgb-d slam in dynamic environments using static point weighting. IEEE Robot. Autom. Lett. **2**(4), 2263–2270 (2017). https://doi.org/10.1109/TMC.2019.2944829
5. Zhang, T., Zhang, H., Li, Y., Nakamura, Y., Zhang, L.: Flowfusion: dynamic dense rgb-d slam based on optical flow. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 7322–7328 (2020). IEEE. https://doi.org/10.1109/ICRA40945.2020.9197349
6. Cui, L., Ma, C.: Sof-slam: a semantic visual slam for dynamic environments. IEEE Access **7**, 166528–166539 (2019). https://doi.org/10.1109/ACCESS.2019.2952161
7. Yu, C., Liu, Z., Liu, X.-J., Xie, F., Yang, Y., Wei, Q., Fei, Q.: Ds-slam: a semantic visual slam towards dynamic environments. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1168–1174 (2018). IEEE. https://doi.org/10.1109/IROS.2018.8593691
8. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(12), 2481–2495 (2017). https://doi.org/10.1109/TPAMI.2016.2644615
9. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
10. Zhong, Y., Hu, S., Huang, G., Bai, L., Li, Q.: Wf-slam: a robust vslam for dynamic scenarios via weighted features. IEEE Sens. J. (2022). https://doi.org/10.1109/JSEN.2022.3169340
11. Soares, J.C.V., Gattass, M., Meggiolaro, M.A.: Crowd-slam: visual slam towards crowded environments using object detection. J Intell. Robot. Syst. **102**(2), 1–16 (2021)
12. Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv:2004.10934 (2020)
13. Shao, S., Zhao, Z., Li, B., Xiao, T., Yu, G., Zhang, X., Sun, J.: Crowdhuman: a benchmark for detecting human in a crowd. arXiv:1805.00123 (2018). https://doi.org/10.48550/arXiv.1805.00123
14. Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K., Leal-Taixé, L.: Mot20: A benchmark for multi object tracking in crowded scenes. arXiv:2003.09003 (2020). https://doi.org/10.48550/arXiv.2003.09003
15. Torr, P.H., Zisserman, A.: Mlesac: a new robust estimator with application to estimating image geometry. Comput. Vis. Image Underst. **78**(1), 138–156 (2000)
16. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 573–580 (2012). IEEE. https://doi.org/10.1109/IROS.2012.6385773
17. Palazzolo, E., Behley, J., Lottes, P., Giguere, P., Stachniss, C.: Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 7855–7862 (2019). IEEE. https://doi.org/10.1109/IROS40897.2019.8967590
18. Liu, Y., Miura, J.: Rds-slam: real-time dynamic slam using semantic segmentation methods. IEEE Access **9**, 23772–23785 (2021). https://doi.org/10.1109/ACCESS.2021.3050617
19. Sun, L., Wei, J., Su, S., Wu, P.: Solo-slam: a parallel semantic slam algorithm for dynamic scenes. Sensors **22**(18), 6977 (2022). https://doi.org/10.3390/s22186977

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

Jianfeng Li is currently pursuing a Master of Science program in Electronic Information at Beijing Information Science & Technology University, Beijing, China, with an expected completion date in 2024. His research focuses on SLAM in dynamic environments, as well as navigation guidance and control.

Juan Dai received the M.S. degree in Fundamental Mathematics from Anhui University, China, in 2009 and the Ph.D. degree in Control Science and Engineering from Beijing Institute of Technology, China, in 2016. During 2016-2018, she was a Postdoctoral Research Associate with the School of Aerospace Engineering, Beijing Institute of Technology, China. She is currently an Associate Professor with the Beijing Information Science and Technology University, Beijing, China. Her current research interests are in the fields of intelligent control systems, autonomous navigation guidance and control, active disturbance rejection control.

Zhong Su received the Ph.D. degree from the Beijing Vacuum Electronics Research Institute, Beijing, China, in 1998. He is currently a Professor with the Beijing Information Science and Technology University, Beijing, China. His current research interests include control, inertial devices, novel gyro sensors, and integrated navigation.

Cui Zhu received the B.S. degree in Automation from China University of Geosciences (Wuhan), China, in 2005 and the Ph.D. degree in Control Science and Engineering from Beijing Institute of Technology, China, in 2014. She is currently an Associate Professor with Beijing Information Science and Technology University, Beijing, China. Her current research interests are in the fields of networked state estimation, multi-sensor information fusion and wireless sensor networks.