Optimized Feature Points and Keyframe Methods for VSLAM in High-Dynamic Indoor Environments

[a]Zhuhua Hu, [a]Wenlu Qi, [a]Kunkun Ding, [a]Hao Qi, [a]Yaochi Zhao*, [b]Xuebo Zhang, [c]Mingfeng Wang

Abstract—**VSLAM is one of the key technologies for indoor mobile robots, used to perceive the surrounding environment, achieve accurate positioning and mapping. However, traditional VSLAM algorithms based on the assumption of a static environment still face certain challenges. The movement, occlusion, and appearance changes of dynamic objects can lead to feature point-matching errors, making data association difficult and causing biases in motion estimation. In order to address this challenge, this paper proposes a dynamic feature point removal method and a closed-loop detection method for high dynamic scenes, aiming to effectively improve the robustness and positioning accuracy in dynamic environments. First, the YOLOv7-tiny object detection network and LK optical flow algorithm are combined to detect the dynamic area, and the adaptive threshold keyframe selection method is adopted to solve the problem of poor quality of keyframe caused by the existing heuristic threshold selection method. Then, this paper proposes a dynamic keyframe sequence creation method based on the angle difference between keyframes, which reduces the workload of loop back detection and accelerates the efficiency of loop back detection in the system. Next, the ParC_NetVLAD image matching algorithm is proposed. In this paper, ConvNeXt-Tiny network is used for feature extraction of images, and ParC-Net network and CBAM attention mechanism are added to the feature extraction network. Finally, NetVLAD is used to cluster the extracted local features to obtain global features that can represent images. Experiments are conducted on public TUM RGB-D datasets and in real-world situations. The proposed algorithm reduces the ATE (Absolute Trajectory Error) by 96.4% and the RPE (Relative Trajectory Error) by 82.8% on average in highly dynamic scenarios. In the Pittsburgh30k dataset, the average accuracy of loop closure detection has been improved by 2.6%.**

Index Terms—**Visual SLAM, Dynamic scene, ORB-SLAM3, YOLOv7-tiny, Keyframe sequences, ParC_NetVLAD**

~~I. Introduction~~

I. Introduction

Nowadays, the technologies of Simultaneous Localization and Mapping (SLAM) have been widely studied because it allows for the instant creation of pose estimation of maps and sensors in unknown environments. Visual SLAM technology plays an important role in this area, for which the camera sensor has the advantages of low cost and rich image information. It has become a research hotspot [1]. However, the assumption that traditional visual SLAM is based on static environments limits its application scenarios. Although the RANSAC (Random Sample Consensus) algorithm can identify the feature points on dynamic objects as outliers and filter them, it is limited to low-dynamic scenes with fewer dynamic elements [2]. In highly dynamic scenes, when dynamic objects occupy a large area of the image, the feature points extracted by traditional visual SLAM (VSLAM) may be distributed on dynamic objects, resulting in a significant decrease in accuracy. The estimated trajectory will no longer be available. To solve this problem, the study of VSLAM in dynamic scenes has attracted widespread attention and has become a frontier topic in current research[3-4].

a School of Information and Communication Engineering, School of Cyberspace Security (School of Cryptology) Hainan University, Haikou 570228, China.
b College of Artificial Intelligence, and Institute of Robotics and Automatic Information System, Nankai University, Tianjin 300350, China
c Department of Mechanical and Aerospace Engineering, Brunel University London, UB8 3PH London, U.K.
*Corresponding author: Yaochi Zhao, cbze5532@21cn.com, zhyc@hainanu.edu.cn

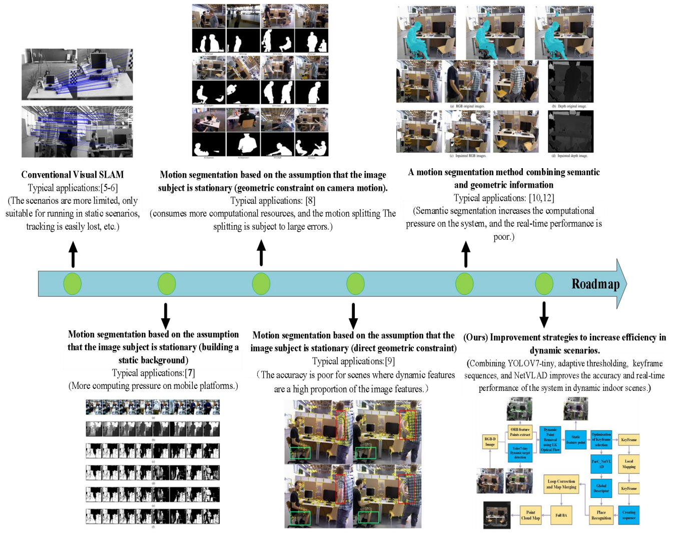

Conventional Visual SLAM
Typical applications:[5-6]
(The scenarios are more limited, only suitable for running in static scenarios, tracking is easily lost, etc.)
Motion segmentation based on the assumption that the image subject is stationary (geometric constraint on camera motion).
Typical applications: [8]
(consumes more computational resources, and the motion splitting The splitting is subject to large errors.)
(a) RGB original images.
(b) Depth original image.
(c) Inpainted RGB images.
(d) Inpainted depth image.
A motion segmentation method combining semantic and geometric information
Typical applications: [10,12]
(Semantic segmentation increases the computational pressure on the system, and the real-time performance is poor.)
Roadmap
Motion segmentation based on the assumption that the image subject is stationary (building a static background)
Typical applications:[7]
(More computing pressure on mobile platforms.)
Motion segmentation based on the assumption that the image subject is stationary (direct geometric constraint)
Typical applications:[9]
(The accuracy is poor for scenes where dynamic features are a high proportion of the image features.)
(Ours) Improvement strategies to increase efficiency in dynamic scenarios.
(Combining YOLOV7-tiny, adaptive thresholding, keyframe sequences, and NetVLAD improves the accuracy and real-time performance of the system in dynamic indoor scenes.)
RGB-D Image
ORB feature Points extract
Yolov7-tiny Dynamic target detection
Dynamic Point Removal using LK Optical Flow
Static feature point
Optimisation of Keyframe selection
KeyFrame
ParC_NetVLAD
Local Mapping
Global Descriptor
KeyFrame
Loop Correction and Map Merging
Point Cloud Map
Full BA
Place Recognition
Creating sequence

Fig. 1. Roadmap of dynamic SLAM.

II. Related Work

A. Research Motivation

In 1986, SmithSelf and Cheeseman first proposed SLAM technology, which has been developed for more than 30 years. In 2017, ORB-SLAM2 [5] was proposed as a comprehensive SLAM solution that supports multiple types of cameras and includes features such as map fusion, loop closing, and repositioning. In 2021, Campos [6] et al. proposed ORB-SLAM3, which is a visual SLAM algorithm that supports multiple cameras. This algorithm optimized many aspects, such as map initialization, repositioning, loop detection, keyframe selection, map construction, and sensor support, and has excellent performance in terms of running speed, tracking results, and mapping accuracy.

The above SLAM algorithms assume that the environment where the robot is located is stationary. Therefore, they cannot recognize complex and changing real scenes, especially when moving objects appear in the scene, and the accuracy and precision of positioning and mapping will be greatly reduced. First, motion segmentation methods based on the assumption of stationary image foreground mainly rely on the hypothesis that most of the features in the image are static and differentiate between static and moving feature points in the image sequence based on the geometric constraints of feature points. Kim et al. [7] constructed a static background based on the initial depth change between images and filtered out moving objects that were inconsistent with the static background in subsequent image sequences, but it was incomplete in filtering out pedestrians. Then, Sun et al. [8] proposed a motion removal method based on RGB-D data as a preprocessing module for filtering moving objects, which is one of the pioneering works in the field of dynamic simultaneous localization and mapping. The GMSK-SLAM system proposed by Wei et al. [9] combines grid-based motion statistical feature matching method and K-means clustering method to detect dynamic regions.

With the significant progress in deep learning for image processing in recent years, motion segmentation methods based on prior semantic information have been applied in dynamic SLAM systems. First, DS-SLAM [10] combines the SegNet [11] semantic segmentation network with conventional SLAM to remove the influence of individual movement in the environment. However, this framework significantly increases the computational pressure of the system by performing semantic segmentation for each frame of the image and cannot achieve real-time performance. Then, the DynaSLAM [12] algorithm combines the MaskRCNN [13] semantic segmentation network with multi-view geometry and uses the consistency of depth information to remove dynamic feature points caused by moving objects. However, this system does not optimize segmentation efficiency. Shao et al. [14] proposed a convolutional network with an improved Faster R-CNN to act as a semantic filter, which filters out patches with low-level semantic labels. The remaining high-level semantic information is then used for exact matching. To address the poor real-time performance of semantic methods in dynamic scenes, Gaurav et al. [15] proposed an approach that utilizes only keyframes to extract semantics, thereby reducing computational overhead. Specifically, the authors extracted semantics only on keyframes where the image content changed significantly. The evolution of SLAM algorithms based on dynamic scenes is shown in Figure 1.

In summary, most existing visual SLAM systems generally suffer from poor robustness or real-time performance in highly dynamic environments. Previous research has focused on improving the accuracy of localization in visual SLAM algorithms, and the processing speed of these algorithms is constrained by the computational demands of their models. There has been little research on improving the real-time performance of visual SLAM systems in depth. Therefore, simultaneously improving the robustness and real-time performance of visual SLAM in indoor high-dynamic scenes is currently a hot issue.

B. Key Contributions

The main contributions of this paper are summarized below:

1. In the front-end part, the innovative use of lightweight target detection network YOLOv7-tiny combined with LK optical flow algorithm to remove the dynamic feature points in the detection frame, and based on the camera observation model, an adaptive threshold algorithm is added to solve the problems of poor keyframe quality of the original heuristic threshold selection and

ensure that the number of keyframe feature points selected by the algorithm is sufficient and evenly distributed.

2. Based on the bag-of-words model in the back-end part, we dynamically create image sequences according to the angle differences between keyframes, replace the previous method of calculating a single BoW vector with the method of calculating the entire sequence's BoW vector, and then match the sequence. This reduces the workload and time consumption of loop detection in SLAM.

3. The ParC_NetVLAD image matching algorithm is proposed, which uses the ConvNeXt-Tiny network to extract features from images, and adds the ParC-Net network to the feature extraction network. Then, in order to extract the channel and spatial information of the feature map, the CBAM attention mechanism is added to the feature extraction network. Finally, NetVLAD is used to cluster the extracted local features to obtain the global features that can represent the image.

4. By verifying the innovative idea, the experimental results show that in the TUM dynamic scene dataset, compared with ORB-SLAM3, the positioning accuracy of our system is improved. The ATE of the proposed algorithm is reduced by 96.4% on average, and the RPE is reduced by 82.8% on average. The Pittsburgh dataset is tested, and the comparison experiments are carried out with NetVLAD, ResNet50, and other neural networks. The experimental results show that the image-matching algorithm proposed in this paper has a higher accuracy. At the same time, the depth camera Astra Pro is tested on the actual dynamic scene, and the running trajectory of our system is more accurate compared with ORB-SLAM3. The algorithm in this paper has been improved in terms of positioning accuracy and real-time performance.

C. Paper Organization

The arrangement of this article is as follows. The opening section of this paper outlines the significance of the research work, related work, and challenges, as well as the main contributions of this paper. Section 2 introduces a visual SLAM system based on YOLOv7-tiny and NetVLAD. Section 3 provides experimental verification of the proposed technical scheme and analyses and discusses the experimental results. Section 4 concludes the whole paper and further points out future directions.

III. Proposed Methods

In dynamic environments, the ORB-SLAM3 algorithm is affected by moving objects, leading to a decrease in positioning accuracy and poor robustness [16-17]. To address this issue, we introduce the YOLOv7-tiny object detection algorithm into the front-end of the ORB-SLAM3 system to simultaneously detect targets in the input image and extract feature points [18]. After obtaining the semantic information in the image, the LK optical flow algorithm is used to determine the dynamic objects present in the image. Based on the target detection results, the dynamic ORB feature points are removed, and only static feature points are retained for pose calculation, thus improving the positioning accuracy of the visual SLAM system [19]. At the same time, an adaptive threshold method is used to optimize keyframe selection for higher quality keyframes. Then, we introduce a serialization process for keyframes before loop detection, combining the BoWs of several consecutive frames into one complete BoW. Based on this sequence (several consecutive frames), we perform loop closure detection to reduce the time required for SLAM loop detection. Next, the improved ConvNeXt network is used to extract features from the image,and the ParC-Net network and CBAM attention mechanism are added to the feature extraction network. Then,NetVLAD is used to cluster the extracted local features to obtain global features that can represent the image. The overall flowchart of the system is shown in Fig. 2, where the blue boxes are improved in this paper.

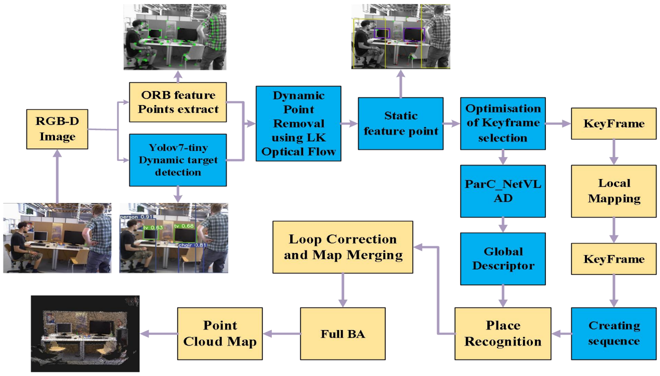


Fig. 2. Overview of our algorithm.

A. Dynamic feature point culling based on YOLOv7-tiny

YOLOv7 [20] is a deep learning-based object detection algorithm with high speed, accuracy, and efficiency, making it suitable for various object detection tasks in different scenarios. In this paper, YOLOv7-tiny is used as the detection model, which mainly consists of four parts: Input, Backbone, Neck, and Prediction. Figure 3 shows the front-end feature point optimization process.

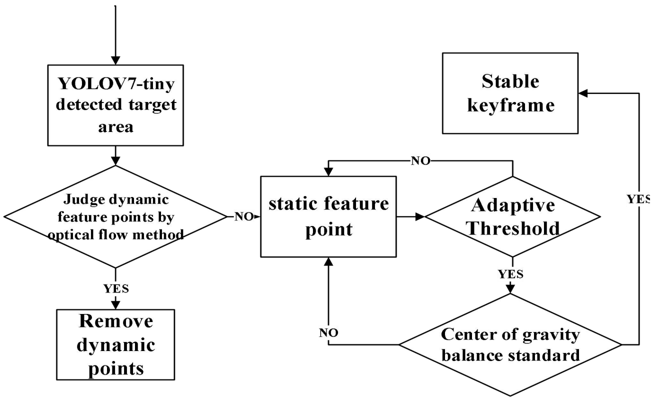

YOLOV7-tiny detected target area
Judge dynamic feature points by optical flow method
YES
Remove dynamic points
NO
static feature point
NO
Adaptive Threshold
YES
Center of gravity balance standard
NO
YES
Stable keyframe

Fig. 3. Flowchart of front-end feature point optimization.

Elimination of dynamic feature points using YOLOv7-tiny. In the dynamic object prediction box generated based on object detection, the dynamic feature points are determined by a priori knowledge, and the elimination process is described as follows:

In the tracking thread of ORB-SLAM3, the feature points are denoted as F_k, and k denotes the k-th frame processed by the system. After being processed by YOLOv7-tiny and the dynamic recognition algorithm, the recognized dynamic objects are denoted as D_k. Removing the dynamic feature points from all the feature points extracted by the ORB-SLAM3 algorithm gives the static feature points P_{k}.

To cull dynamic objects, this study introduces the YOLOv7-tiny target detection algorithm at the front end of the system to detect targets in the input image. The LK optical flow algorithm is then utilized to determine the dynamic objects in the target detection frame [21-24]. The ORB feature point located at *(x, y)* in the image at time *t* can be expressed as *I(x, y, t)*. Based on the grayscale invariance assumption of the optical flow method, [25] derives that:

$$\begin{bmatrix} I_x I_y \end{bmatrix}_k \begin{bmatrix} u \\ v \end{bmatrix} = -I_{tk}, k = 1, \dots, w^2 \qquad (1)$$

where u and I_x are the velocity and gradient of the motion of the feature point on the X-axis, respectively. v and I_y are the velocity and gradient on the Y-axis, respectively. I_t is the amount of variation of the feature point's grayscale against time. The symbol w represents the size of the window being set. If the optical flow of a feature point is greater than the threshold ε_{th}, the point is considered a dynamic point.

The static feature points are optimized for keyframes using the camera observation model proposed by Azimi et al. [26]. The adaptive threshold is determined based on the number of points where the line of sight changes, allowing for an accurate modeling and selection of the current frame as a key frame. In addition to satisfying adaptive thresholding, the stability of the frame must also be considered, with the center of gravity threshold used to determine if the points are evenly distributed throughout the image. After the image enters the SLAM system, it will be divided into 3×3 grids, and a 3×3 matrix will be generated. In each grid, the points with a conical area change greater than 30 degrees are counted as effective points, and this angle has been determined by previous experiments [27].

Suppose that the coordinates of each pixel of the image are x_i, the pixel value is p_i, and the direction coordinate of the centre of gravity is denoted as x.

$$\sum_{i=1}^{n} p_i (x_i - x) = 0 \qquad (2)$$

Then the center of gravity in the x,y direction is:

$$\begin{cases} x = \dfrac{\sum_{i=1}^{n} p_i x_i}{\sum_{i=1}^{n} p_i} \\ \mathrm{y} = \dfrac{\sum_{i=1}^{n} p_i y_i}{\sum_{i=1}^{n} p_i} \end{cases} \qquad (3)$$

The centre of gravity of the matrix serves as the threshold T_1,

$$T_1 = \sqrt{\mathrm{x}^2 + y^2} \qquad (4)$$

T_2 Defined as:

$$T_2 = \frac{\sqrt{(c_{\max_1} - c_{\max_2})^2 + (r_{\max_1} - r_{\max_2})^2} * M_2}{2\sqrt{2} M_1} \qquad (5)$$

where M_1 and M_2 are the two largest numbers within the matrix, and c and r are the numbers of rows and columns, respectively. If $\mathrm{T}_2 > \mathrm{T}_1$, the feature point distribution of the current frame is considered more appropriate, and the frame has sufficient stability.

B. The dynamic creation method of keyframe sequences for optimizing closed-loop detection

Based on the research of Bampis et al. [28], we create a sequence of keyframes in ORB-SLAM3 to recognize keyframes generated by the local mapping thread as a sequence with common features. R_{cw} is the rotation matrix that describes the rotation from the world coordinate system to the camera coordinate system. From ORB-SLAM3, we can derive the rotation matrix R_{cw2} of the current keyframe k_2 and the rotation matrix R_{cw1} of the previous keyframe k_1 and then calculate the rotation matrix R_{cw}^{r1} between the two frames.

$$R_{cw}^{r_1}=R_{cw2}\times(R_{cw1})^{-1} \tag{6}$$

From the rotation matrix, we can find the angle of rotation between the two θ_1:

$$\theta_1=\arccos\frac{tr(R_{cw}^{r_1})-1}{2} \tag{7}$$

where tr(X) is the sum of the diagonal elements of matrix X.

Similarly, you can find the rotation matrix R_{cw}^{r2} between a keyframe k_2 and its next keyframe k_3:

$$R_{cw}^{r_2}=R_{cw3}\times(R_{cw2})^{-1} \tag{8}$$

From the rotation matrix, we can find the angle of rotation between the two θ_2:

$$\theta_2=\arccos\frac{tr(R_{cw}^{r_2})-1}{2} \tag{9}$$

Meanwhile, we set an angle threshold θ_T =0.1. If $\theta_1>\theta_T$ and $\theta_2<\theta_T$, then no new sequence will be created, and new keyframes will continue to be added to the current sequence. If $\theta_1<\theta_T$ and $\theta_2>\theta_T$, then the current sequence ends, and a new sequence will be created. No new sequence will be created in all other cases. Additionally, to prevent sequences from being too long or too short, we set a length threshold L_T =15 and a sequence length L_S. No new sequence will be created when the sequence length L_S is less than 2, and the current sequence ends and a new sequence is created when the sequence length L_S exceeds the length threshold L_T. The algorithm for creating new sequences is shown in Algorithm 1.

Algorithm 1: Creating sequence algorithm

```
input: KeyFrame $K_F$, Length of the keyframe list $L_K$, Length of current sequence $L_S$
output: Results after verification of new sequences
Initialization $\theta_T = 0.1$, $L_T$=15, i=0
for i←0 to $L_K$ do
    $\theta_{i+1}$ = CalAngle($K_{F(i)}$, $K_{F(i+1)}$)
     //Calculate the angle of rotation between two keyframes
    if $L_S$<2 then
        newsequence=0
    end if
    if $L_S$>$L_T$ then
        newsequence=1
    end if
    if $\theta_j$>$\theta_T$ and $\theta_{j+1}$<$\theta_T$ then
        newsequence=0
    end if
    if $\theta_j < \theta_T$ and $\theta_{j+1}$>$\theta_T$ then
        newsequence=1
    end if
            // For all other remaining cases, newsequence=0
end for
```

C. Improved NetVLAD-based loop closure detection algorithm

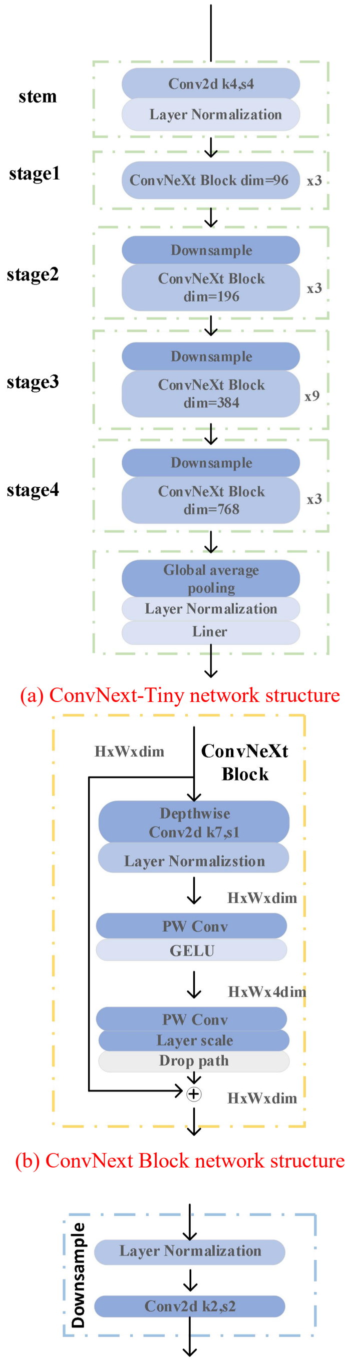
Although bag-of-words model-based closed-loop detection can achieve certain results in some cases, this method is mainly based on artificially designed features, which are difficult to extract in complex environments. Therefore,the bag-of-words model-based closed-loop detection algorithm is not effective in complex environments,which seriously limits the development of visual SLAM [29].

The closed-loop detection based on deep learning is essentially image re-identification [30], and the key technology is image matching technology. Based on this core, this paper proposes the ParC_NetVLAD image-matching algorithm. Firstly, to extract more representative features of the image, the ConvNeXt-Tiny [31] network (hereinafter referred to as ConvNeXt) is used for feature extraction of the image. Secondly, to extract the location information features of the image, the ParC-Net [32] network is added to the feature extraction network. Then, to extract the channel and spatial information of the feature map, the CBAM [33] attention mechanism is added to the feature extraction network. Finally, NetVLAD [34] is used to cluster the local features extracted to obtain the global features that can represent the image.

1) IMPROVED CONVNEXT NETWORK

ConvNeXt network was first used in image classification, and its accuracy has exceeded Swin Transformer [35] under the same FLOPs. Its structure is shown in Figure 4.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI: 10.1109/TITS.2024.3520177, IEEE Transactions on Intelligent Transportation Systems



(a) ConvNext-Tiny network structure

(b) ConvNext Block network structure

(c) Downsample network structure

Fig. 4. ConvNeXt-Tiny.

In order to better apply the ConvNeXt network in closed-loop detection, this paper removes the final global average pooling layer, regularization layer, and Liner layer. In addition, in order to extract the location information in image features, the ParC-Net network is introduced. In order to ensure the advantages of the ConvNeXt Block module and better utilize the ability of the ParC-Net module to focus on location information, ParC-Net is added to stage 3 and stage 4 of the ConvNeXt network. ParC is added to the last three ConvNeXt Block modules in stage 3 and the last ConvNeXt Block module in stage 4. The structure of the ParC-ConvNeXt Block module obtained after replacement is shown in Figure 5.

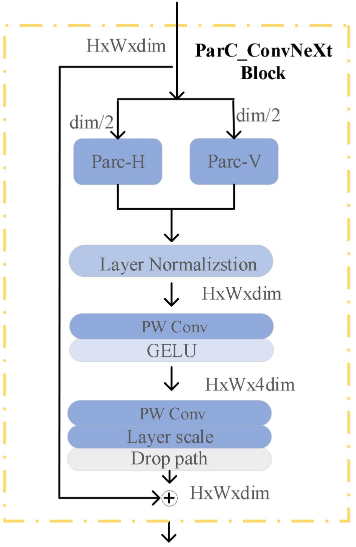


Fig. 5. ParC_ConvNeXt Block.

At this time, the information on the features extracted by the improved ConvNeXt network still cannot meet the needs. In order to better extract features, this paper adds the CBAM attention mechanism after the improved ParC_ConvNeXt network. CBAM attention mechanism can adjust the receptive field of the convolutional neural network according to the image content so that the model can better capture the important information in the image. It can adaptively adjust the feature map at different levels so that the model can better distinguish between important features and noise. In addition, the CBAM attention mechanism can also reduce overfitting and improve the generalization ability of the model.

2) CLOSED-LOOP DETECTION ALGORITHM BASED ON IMPROVED NETVLAD NETWORK

Deep neural networks can learn deeper features of the image by convolution operation, and can accurately reflect image information. It is guaranteed that accurate image matching can still be achieved in complex environments so that closed-loop detection can have better effects. This paper studies the NetVLAD network, and on this basis proposes an improved ParC_NetVLAD, which is used for closed-loop detection. The NetVLAD network is improved on the basis of VLAD (Vector of Locally Aggregated Descriptors) [36]. The idea of VLAD is to gather all the local descriptors (such as SIFT, SURF, ORB, etc.) in an image and form a vector, which is used to represent the global descriptor of the whole image.

The VGG19 convolution network is used in the NetVLAD network to extract image features. Although the extracted image features can contain a lot of information about the image, the image features extracted by the VGG19 network lack position information, and the deeper features in the image cannot be better used. In addition, the VGG19 network has a large number of parameters in the training process, which is easy to overfit the training data, resulting in the decline of model performance. In this paper, aiming at the shortcomings of VGG19 in image feature extraction, the improved ConvNeXt network is proposed to extract image features. Finally, the extracted features are processed by the NetVLAD module to generate the required K*D as features. The improved network is shown in Figure 6.

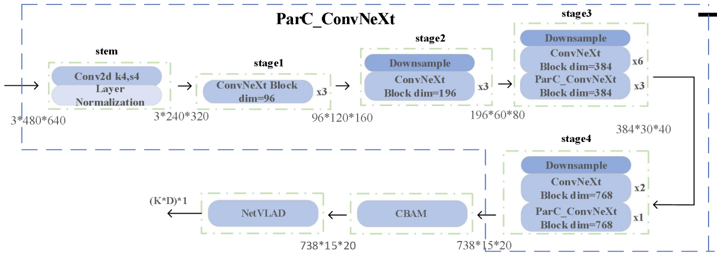


Fig. 6. Network structure of ParC_ConvNeXtT_NetVLAD.

In the whole improved network, the input image with dimension of 3*480*640 is input. The image is output with feature dimension of 738*15*20 after ParC_ConvNeXt module. The feature map is sent to CBAM module to extract channel and space information and keep the dimension of the feature map unchanged. Then the feature map is sent to NetVLAD module for dimension reduction, so that the feature map becomes the required K*D dimension feature, i.e. 64*738.

IV. Experimental results and analysis

A. Dataset and Experimental Environment

In this experiment, we mainly used the TUM [37] dataset, Pittsburgh30K [38] dataset, and Tokyo247 [39] dataset. The TUM dataset is obtained by collecting RGB-D and monocular image sequences and corresponding motion capture live trajectories in indoor and outdoor environments. The dataset is challenging due to the presence of dynamic objects, changing lighting conditions, and large camera motion, making it a good test platform for evaluating the robustness and accuracy of SLAM algorithms. Pittsburgh30K is a subset dataset created by the Department of Computer Science at Carnegie Mellon University in 2012, containing more than 30,000 high-resolution street-view images. Tokyo247 is a large-scale street-view image dataset created by Waseda University and the University of Tokyo, containing more than 240,000 images. These datasets are widely used in computer vision tasks and urban planning research and are valuable in human activity recognition, autonomous driving, and intelligent transportation systems.

The experimental environment is shown in Table 1:

Hardware/Software component	Specification
Operating system	Ubuntu18.04
GPU	RTX 3090(24GB)
CPU	12 vCPU Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50GHz
PyTorch	1.8.0
Python	3.7

Table 1. Experimental environment.

B. Evaluation metrics

In this paper, the trajectory accuracy is evaluated to verify the optimization degree of our method. The pose estimation error analysis experiment uses the evo tool, as well as the evaluate_ate and evaluate_rpe tools on the TUM website to test and compare the camera pose estimated by the ORB-SLAM3 system CameraTrajectory.txt with the real pose given by the dataset groundtruth.txt. The evaluation parameters mainly adopt relative pose error (RPE) and absolute trajectory error (ATE) [37].

The comparison with the error of ORB-SLAM3 and the relative lift rate η is calculated as follows:

$$\eta = \frac{orbslam3 - ours}{orbslam3} \times 100\% \qquad (10)$$

In the loop detection experiment, the features of each query image are first clustered, and then the L2 distance is compared with the features within the class. The L2 distance calculation formula is shown as follows.

$$d_2 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \qquad (11)$$

Then select the 20 images with the highest similarity and sort them, and finally compare whether the 20 images contain the query image scene in order. When comparing the similarity, we set N=[1,5,10,20]. If the first image is a correct match, then add 1 to the counter under N=1. If the first five images have a correct match, then add 1 to the counter under N=5. Similarly, for N=10 and N=20, the same operation is carried out. After all the query images are matched, the value of the counter under each N is counted.

Then divide it with the total number of query images to find the accuracy.

C. Experimental results analysis

TUM is a dataset for indoor dynamic environments. The results will be compared with ORB-SLAM3 [6], Dyna SLAM [12], DS-SLAM [10], SOF-SLAM [40], and OFM-SLAM [41]. Table 2 compares the ATE with ORB-SLAM3, and Table 3 and Table 4 compare the RPE. Four metrics were measured for the performance of the VSLAM system, namely Standard Deviation (STD), Root Mean Square Error (RMSE), Median Error (Median), and Mean Error (Mean). In dynamic scenarios, our performance is somewhat improved compared to most algorithms. Visual interference can be better suppressed when the object is moving faster, resulting in more accurate localization and mapping. However, in the case of slowly moving objects, the optimization strategy used in the algorithm has only a minor impact on system accuracy, resulting in a slight improvement over the ORB-SLAM3 algorithm. Therefore, the proposed algorithm can be considered as a promising solution for SLAM in dynamic environments with fast-moving objects.

Table 2. Comparison of absolute trajectory error between ORB-SLAM3 and our algorithm (ATE/unit:m).

Datasets	ORB-SLAM3				Ours				Improvements%			
	Mean	Median	RMSE	STD	Mean	Median	RMSE	STD	Mean	Median	RMSE	STD
walking_static	0.3256	0.3173	0.3598	0.1524	0.0057	0.0051	0.0066	0.0032	98.25%	98.39%	98.17%	97.90%
walking_xyz	0.5534	0.4824	0.6426	0.3251	0.0127	0.0109	0.0151	0.0081	97.71%	97.74%	97.65%	97.51%
walking_rpy	0.7378	0.7168	0.821	0.3558	0.0262	0.0194	0.0405	0.0309	96.45%	97.29%	95.07%	91.32%
walking_half	0.5086	0.4889	0.5466	0.194	0.0169	0.0142	0.0223	0.0145	96.68%	97.10%	95.92%	92.53%
sitting_static	0.0067	0.0060	0.0077	0.0038	0.0054	0.0046	0.0062	0.0031	19.40%	23.33%	19.48%	18.42%

Table 3. Results of translational relative pose error (RPE/unit：m).

Datasets	ORB-SLAM3				Ours				Improvements%			
	Mean	Median	RMSE	STD	Mean	Median	RMSE	STD	Mean	Median	RMSE	STD
walking_static	0.1021	0.0213	0.2225	0.1977	0.0099	0.0081	0.0124	0.0074	90.30%	61.97%	94.43%	96.26%
walking_xyz	0.2621	0.1643	0.3578	0.2435	0.0183	0.0158	0.0212	0.0107	93.02%	90.38%	94.07%	95.61%
walking_rpy	0.2701	0.1427	0.3903	0.2817	0.0456	0.0283	0.0828	0.0691	83.12%	80.17%	78.79%	75.47%
walking_half	0.1444	0.0556	0.2373	0.1882	0.0253	0.0215	0.0321	0.0197	82.48%	61.33%	86.47%	89.53%
sitting_static	0.0085	0.0076	0.0097	0.0046	0.0068	0.0060	0.0077	0.0037	20.00%	21.05%	20.62%	19.57%

Table 4. Results of rotational relative pose error (RPE/unit：deg).

Datasets	ORB-SLAM3				Ours				Improvements%			
	Mean	Median	RMSE	STD	Mean	Median	RMSE	STD	Mean	Median	RMSE	STD
walking_static	2.6563	2.9369	3.0119	1.4053	0.2571	0.2278	0.2986	0.1518	90.32%	92.24%	90.09%	89.20%
walking_xyz	4.9684	3.1978	6.7628	4.5852	0.5081	0.4164	0.644	0.3956	89.77%	86.98%	90.48%	91.37%
walking_rpy	5.2563	2.6908	7.5570	5.4278	0.6988	0.3264	4.2352	4.1771	86.71%	87.87%	43.96%	23.04%
walking_half	2.8593	1.3501	4.5654	3.5575	0.6397	0.5782	0.7276	0.3465	77.63%	57.17%	84.06%	90.26%
sitting_static	0.2646	0.2528	0.2929	0.1255	0.2390	0.2232	0.2653	0.1151	9.67%	11.71%	9.42%	8.29%

To ensure the effectiveness of each module, we conducted ablation experiments on YOLOv7-tiny and the adaptive threshold module. The experimental results are shown in Table 5 (✓denotes the existence of this module, and × denotes the removal of this module), and the evaluation criterion is the RMSE value. Table 5 clearly shows the impact of each module. From the data results, it can be concluded that using both modules simultaneously achieve the minimum root mean square error for the algorithm.

Table 5. Ablation experiment ATE.

YOLOv7-tiny	Adaptive thresholds	walking_xyz	walking_half	walking_static	walking_rpy
✓	✓	0.0151	0.0223	0.0066	0.0405
✓	×	0.0158	0.0239	0.0113	0.0535
×	✓	0.2988	0.1930	0.0274	0.3276
×	×	0.6426	0.5466	0.3598	0.821

In Table 6, our algorithm is compared with other SLAM methods in dynamic environments. From Table 6, it can be seen that the highest positioning accuracy is achieved by DynaSLAM and our algorithm. Since DynaSLAM uses a MaskRCNN semantic segmentation network that performs per-pixel execution, the recognition precision is higher. Based on YOLOV7-tiny, our visual SLAM strategy can quickly detect and track moving objects. Although it has slightly lower accuracy than the DynaSLAM algorithm, it has a significant advantage in speed, making it suitable for real-time scenarios that require fast processing. We performed time tests on our algorithm and DynaSLAM algorithm on the w_xyz dataset, and Table 7 lists the processing time per frame for both our algorithm and the DynaSLAM algorithm. Compared with the DynaSLAM algorithm, it shows superior performance in terms of real-time processing speed.

Table 6.The absolute trajectory error of different algorithms(ATE).

Datasets	ORB-SLAM3 [6]	DynaSLAM[12]	DS-SLAM [10]	SOF-SLAM[40]	OFM-SLAM[41]	**Ours**
walking_static	0.3598	0.0073	0.0081	0.0070	0.0081	**0.0066**
walking_xyz	0.6426	**0.0131**	0.0247	0.0180	0.0168	0.0151
walking_rpy	0.8210	**0.0279**	0.4442	0.0270	0.0326	0.0405
walking_half	0.5466	**0.0174**	0.0303	0.0290	0.0252	0.0223
sitting_static	0.0077	**0.0058**	0.0065	0.0100	0.0065	0.0062

Table 7.Tracking time comparison (unit ms).

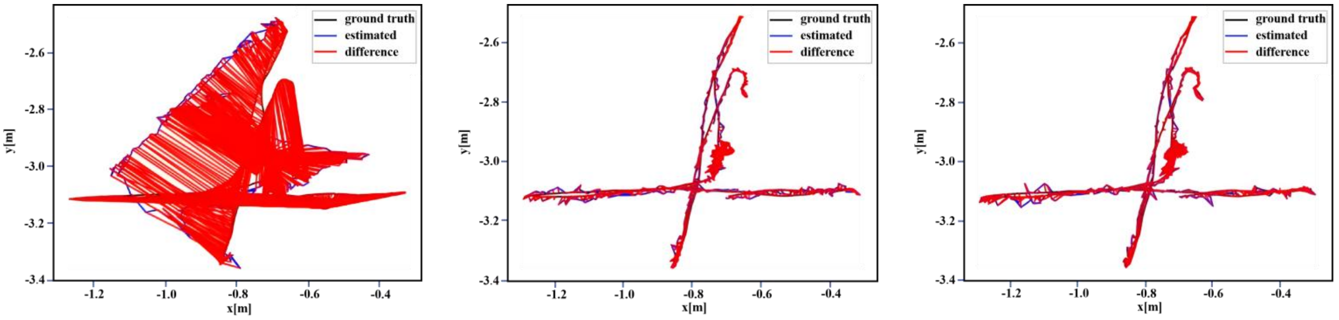
Algorithm	Time
Dyna-SLAM[10]	2452
Ours	29
Improvements	98.78%

Table 8 presents a comparison of the testing time for loop closure detection between using the sequential keyframe method in ORB-SLAM3 and ORB-SLAM3 alone. It can be seen that in datasets with indoor loops, such as fr1_room and fr2_desk, the total time for loop closure detection is relatively reduced when using the sequential keyframe method.

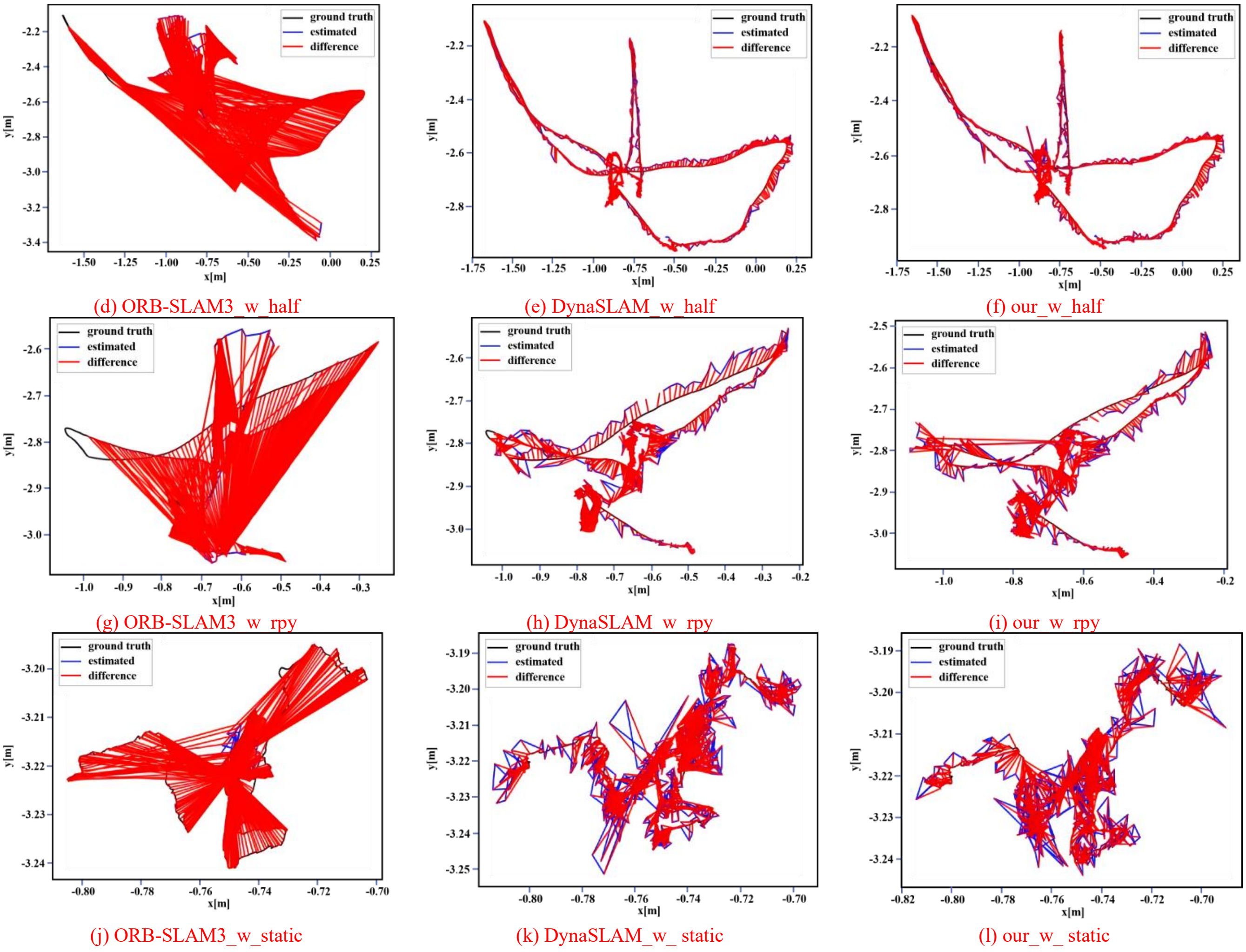
Table 8. Loop Closing total time comparison (unit ms).

Datasets	ORB-SLAM2[5]	ORB-SLAM3	Ours	Improvements
fr1_room	1170	379	346	8.71%
fr2_desk	1584	571	544	4.73%

Figure 7 shows the trajectory and error distribution estimated by ORB-SLAM3, DynaSLAM, and our algorithm. In the figure, the black, blue, and red lines denote the true trajectory, the estimated trajectory, and the error between them. The graphic clearly illustrates the high level of accuracy in the estimated pose by the algorithm presented in this article, which closely matches the ground truth trajectory. Figure 8 shows the error distribution and estimated trajectory of our algorithm's RPE value, which indicates that the algorithm has small errors and that the trajectory is estimated to be very close to the ground truth.



(a) ORB-SLAM3_w_xyz (b) DynaSLAM_w_xyz (c) our_w_xyz

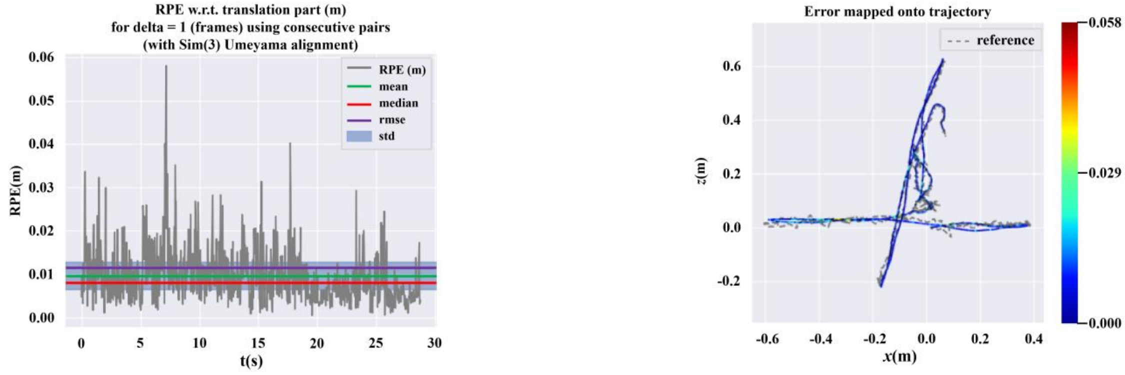


(d) ORB-SLAM3_w_half (e) DynaSLAM_w_half (f) our_w_half

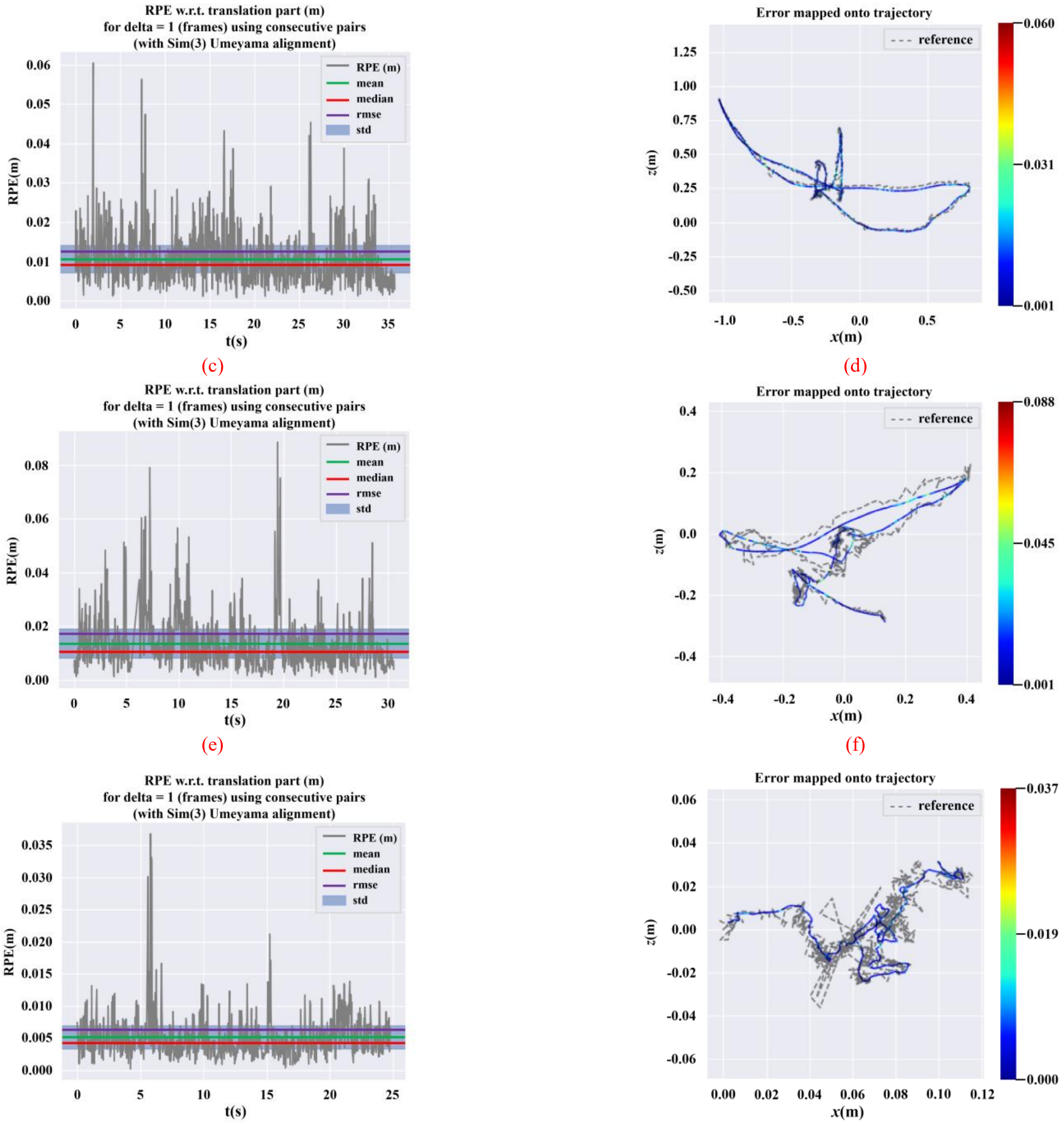
(g) ORB-SLAM3_w_rpy (h) DynaSLAM_w_rpy (i) our_w_rpy

(j) ORB-SLAM3_w_static (k) DynaSLAM_w_ static (l) our_w_ static

Fig. 7. Comparison of estimated trajectories and real trajectories in highly dynamic environments.



(a) (b)

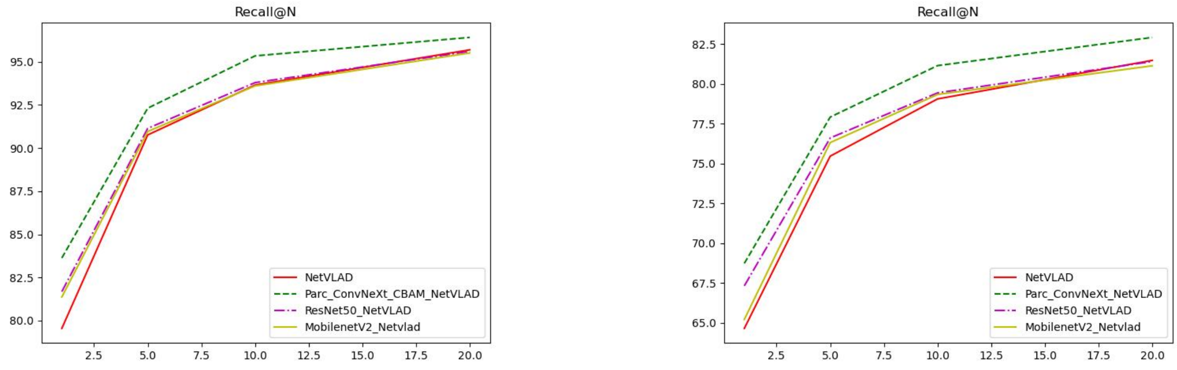


(c) (d)

(e) (f)

(g) (h)

Fig. 8. Relative pose error. (a) (b) walking-xyz. (c) (d) walking-halfsphere. (e) (f) walking-rpy. (g) (h) walking-static.

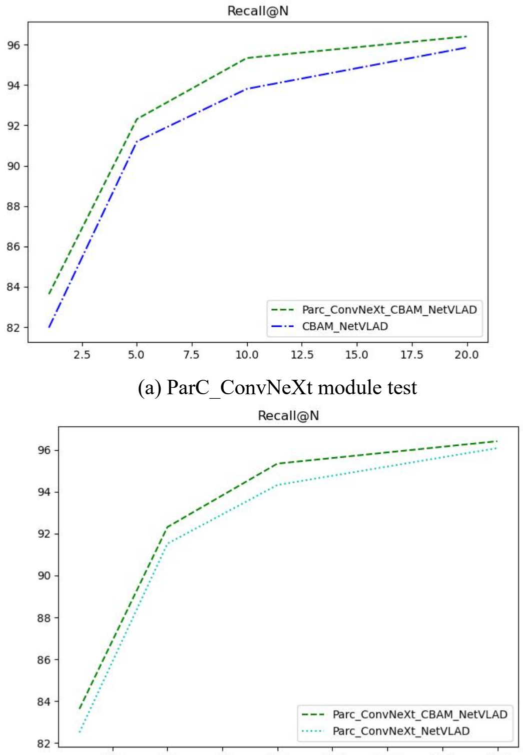


(a) Pittsburgh30k dataset

(b) Tokyo247 dataset

Fig. 9. Experimental results of the dataset.

The improved ParC_NetVLAD algorithm for loop detection is compared with the original algorithm NetVLAD, ResNet50_NetVLAD and other algorithms, and the comparison results are shown in Figure 9. It can be seen from Figure 9 that in the Pittsburgh30k dataset, the improved model has an accuracy of 83.83% when N=1, which is 79.55% higher than the original NetVLAD model. It is 2.15% and 2.46% higher than the ResNet50_NetVLAD model and the MobilenetV2_NetVLAD model, respectively. When N=5, it is 2.55% higher than the original model NetVLAD, 2.17% and 2.35% higher than the ResNet50_NetVLAD model and the MobilenetV2_NetVLAD model, respectively. When N=10, it is 1.69% higher than the original model NetVLAD, 1.55% and 1.55% higher than the ResNet50_NetVLAD model and the MobilenetV2_NetVLAD model, respectively. When N=20, it is 1.72% higher than the original model NetVLAD, 1.81% and 1.9% higher than the ResNet50_NetVLAD model and the MobilenetV2_NetVLAD model, respectively.



(a) ParC_ConvNeXt module test

(b) CBAM module test

Fig. 10. Comparison of ablation experiments.

At the same time, in order to verify the effectiveness of the improved ParC_ConvNeXt model. This paper will use CBAM_NetVLAD network and ParC_ConvNeXt_NetVLAD network to test on the Pittsburgh30k dataset, as shown in Figure 10(a). It can be seen from Figure 10(a) that the accuracy of the ParC_ConvNeXtT module is higher than that without the use of the module. When N=1, it is 1.86% higher; when N=5, it is 2.11% higher; when N=10, it is 1.53% higher; when N=20, it is 1.55% higher. Figure 10(b) is a comparative test to verify the effectiveness of the CBAM attention mechanism module. It can be seen that the network with the CBAM attention mechanism module is 1.6% higher than that without the CBAM attention mechanism module when N=1; when N=5, it is 1.8% higher; when N=10, it is 1.12% higher; when N=20, it is 1.24% higher. In summary, the closed-loop detection based on the improved NetVLAD algorithm studied in this paper has a certain improvement in the accuracy of image matching, which can better improve the closed-loop detection performance of the image scene matching system.

D. Effects of the actual scene

To further validate the effectiveness of our proposed algorithm, experiments were conducted in real-world scenarios using a Jetson Nano mobile robot as the moving platform and an Astra Pro RGB-D camera produced by ORBBEC Corporation as the visual sensor. The appearance and parameters are shown in Figure 11 and Table 9, respectively. The depth measurement part of Astra Pro adopts structured light technology and performs depth ranging through the triangulation principle, showing excellent performance in indoor environments.

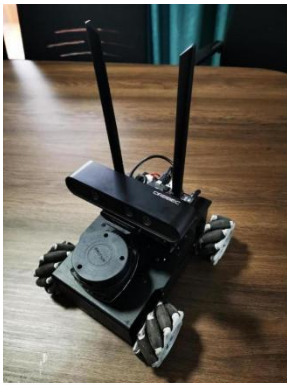


Fig. 11. Mobile robot and Astra Pro camera.

Table 9 Astra Pro parameters.

Astra Pro parameter	
Parameter	Values
Measuring range	0.6m - 8m
Resolution of color image	1280×720@30FPS
Resolution of depth map	1280×1024@7FPS
Precision	(±1-3)mm@1m
Colour FOV	H66.1°and V40.2°
Depth FOV	H58.4°and V45.5°
Delay (ms)	30-45

The experimental site is a rectangular conference room, as depicted in Figure 12. The intelligent robot departs from point A and moves in a clockwise direction of A, B, C, D, and back to A. At the same time, people will walk back and forth at points B and C during the movement, as shown in Figure 13,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TITS.2024.3520177, IEEE Transactions on Intelligent Transportation Systems

allowing for the recognition of dynamic objects in indoor scenes.

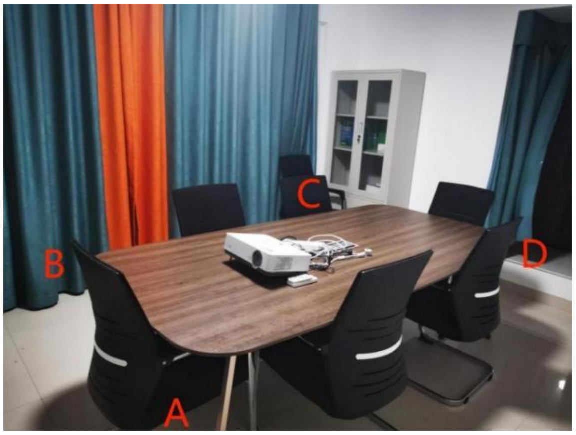
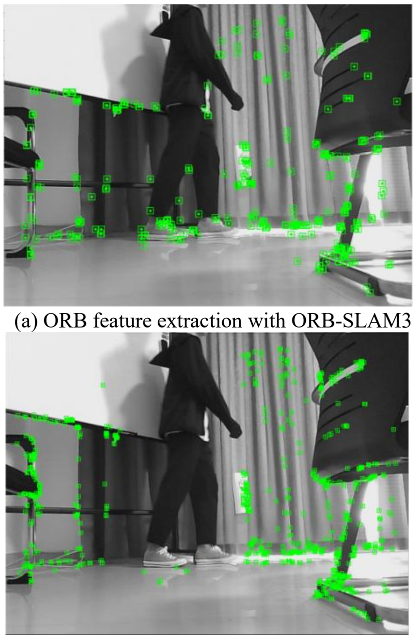


Fig. 12. Experimental site.



(a) ORB feature extraction with ORB-SLAM3

(b) ORB feature extraction with our system

Fig. 13. Comparison of ORB-SLAM3 with ORB feature extraction in our system. This method removes the dynamic features of the walking human body.

Finally, after multiple evaluations in real-world environments, the actual running trajectories of indoor dynamic scenes based on ORB-SLAM3 and our algorithm are displayed in Figure 14. It can be observed that the actual trajectory based on ORB-SLAM3 has a significant deviation under the interference of dynamic objects while the one based on our method is much more accurate and can better close the loop.

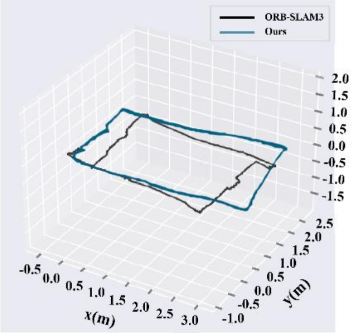


Fig. 14. ORB-SLAM3 compared to the actual operating trajectory of our system.

In the office experiment, we used the camera signal and the corresponding timestamp to determine the trajectory calculation value of the passing point. We set the starting point and the endpoint at the position of point A, and set one point at the position of B, C, and D. In addition, we also added a point in the middle of two points, forming a total of nine passing points.

Table 10. Experimental Results of Trajectory Localization in the Office Environment.

	Ours	ORB-SLAM3	Ground truth
Route 1	(-0.0003,-0.0003)	(-0.0004,-0.0003)	(-0.0001,-0.0005)
Route 2	(-0.0417,0.9185)	(-0.0472,0.9234)	(-0.0591,0.9296)
Route 3	(0.0281,2.0783)	(-0.0391,2.1288)	(0.0436,2.0787)
Route 4	(1.2409,2.0664)	(1.1703,2.1391)	(1.2366,2.0964)
Route 5	(3.0418,2.2329)	(2.9859,2.3148)	(2.9987,2.2875)
Route 6	(3.0636,1.3308)	(2.7756,1.3489)	(3.0198,1.3713)
Route 7	(2.8889,0.4121)	(2.8343,0.3916)	(2.9619,0.3988)
Route 8	(1.9701,0.3202)	(1.9048,0.3062)	(1.9807,0.3010)
Route 9	(0.0681,0.0071)	(0.0384,0.0236)	(0.0724,0.0060)
RMSE	**0.0421**	**0.1056**	—

In this indoor flat experimental environment, the car moves horizontally and the coordinates change very little in the vertical direction, so this paper only focuses on the plane positioning accuracy. To obtain the true value of the trajectory, the passing points of the vehicle were marked with millimeter accuracy through manual measurement during the experiment. In this paper, the two-dimensional coordinates are used to represent the position of the camera and the conversion relationship between the coordinate system where the aligned trajectory is located and the office coordinate system is determined through the calculation of the starting position and direction. In this way, we can calculate the true coordinates of the passing points. The experimental results are shown in Table 10. According to the RMSE of the ATE of the passing points, the positioning accuracy of the algorithm in this paper is significantly improved, which is 60.13% higher than that of ORB-SLAM3.

V. Conclusions

Moving objects may introduce motion blur, occlusion, and environmental appearance changes, which make it difficult for VSLAM systems to maintain accurate and consistent estimations. In this paper, an efficient deep neural network object detection method YOLOv7-tiny algorithm is applied to ORB-SLAM3, and the adaptive threshold method is used to replace the heuristic threshold in ORB-SLAM3 to ensure the quality of key frames selected by the algorithm in dynamic scenes. By filtering dynamic objects in the environment and extracting feature points of static regions, the accuracy of the visual SLAM algorithm in dynamic scenes and the robustness of the system are improved.

Then, in loop detection, keyframes are processed based on the bag of binary features. According to the angle difference between keyframes, the key frame sequence is dynamically divided into key frame sequences to reduce the time consumption of the loop detection part of SLAM. At the same time, the NetVLAD is improved, the ParC_ConvNeXt network is designed to focus on the location information in the image feature map, and the CBAM attention mechanism is added to the network to focus on the feature channel and spatial information. Finally, the obtained image features are sent to the NetVLAD pooling network for dimensionality reduction.

Our algorithm reduces the average ATE by 96.4% and the average RPE by 82.8% in high-dynamic scenes, which is better than most dynamic visual SLAM. In addition, compared to the localization accuracy of DynaSLAM, the time consumption of the tracking thread in our algorithm is reduced by more than 98%. Therefore, our algorithm significantly improves the computational speed while enhancing the localization precision.

However, the performance of our algorithm on the TUM low-dynamic scenes is far inferior to the ORB-SLAM3 algorithm's localization accuracy. This is also a common drawback of many visual SLAM algorithms based on dynamic scenes. To improve the adaptability of algorithms in different environments, more flexible and intelligent methods for detecting different environmental changes need to be developed.

The experimental results on the Pittsburgh30 dataset and Tokyo247 dataset show that the accuracy of the improved algorithm is up to 4.25% higher than that of the other methods when N=1, 2.55% higher when N=5, 1.69% higher when N=10, and 1.9% higher when N=20. In summary, this paper improves the positioning accuracy and real-time performance of visual SLAM in indoor high-dynamic situations.

Future research efforts should focus on optimizing the algorithm model and improving the accuracy of indoor dynamic scene localization. Additionally, it would be worthwhile to explore leveraging semantic information extracted from object detection to build semantic maps, enabling the system to handle higher-level tasks.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (Grant no. 62161010, 62361024), the Key Research and Development Project of Hainan Province (Grant no. ZDYF2022GXJS348 and Grant no. ZDYF2024GXJS021), and the Hainan Province Natural Science Foundation (623RC446). The authors would like to thank the referees for their constructive suggestions.

REFERENCES

[1] Shen X, Chen L, Hu Z, Fu Y, Qi H, Xiang Y, Wu J. A Closed-loop Detection Algorithm for Online Updating of Bag-of-Words Model [C]// The 9th International Conference on Computing and Data Engineering (ICCDE 2023), Haikou, China, January 6-8, 2023.

[2] Cai D, Li R, Hu Z, Lu J, Li S, Zhao Y. A comprehensive overview of core modules in visual slam framework. Neurocomputing, 127760(2024).

[3] Fu Y, Han B, Hu Z, Shen X, Zhao Y. Cbam-slam:A semantic slam based on attention module in dynamic environment[C]// The 6th Asian Conference on Artificial Intelligence Technology(ACAIT 2022), pp. Changzhou, China, December 10-12 2022.

[4] Fu Y, Han B, Hu Z, Shen X, Zhao Y. CBAM-SLAM: A semantic SLAM based on attention module in dynamic environment [C]// The 6th Asian Conference on Artificial Intelligence Technology (ACAIT 2022), Changzhou, China, December 10-12 2022.

[5] Mur-Artal R, Tardós J D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras[J]. IEEE Transactions on Robotics, 2017, 33(5): 1255-1262.

[6] Campos C, Elvira R, Rodríguez J J G, José M. M. M.,Juan D. T. Orb-slam3: An accurate open-source library for visual, visual−inertial, and multimap slam[J]. IEEE Transactions on Robotics, 2021, 37(6): 1874-1890.

[7] Kim D H, Kim J H. Effective background model-based RGB-D dense visual odometry in a dynamic environment[J]. IEEE Transactions on Robotics, 2016, 32(6): 1565-1573.

[8] Sun Y, Liu M, Meng M Q H. Improving RGB-D SLAM in dynamic environments: A motion removal approach[J]. Robotics and Autonomous Systems, 2017, 89: 110-122.

[9] Wei H, Zhang T, Zhang L. GMSK-SLAM: a new RGB-D SLAM method with dynamic areas detection towards dynamic environments[J]. Multimedia Tools and Applications, 2021, 80: 31729-31751.

[10] Yu C, Liu Z, Liu X J, et al. DS-SLAM: A semantic visual SLAM towards dynamic environments[C]//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018: 1168-1174.

[11] Badrinarayanan V, Kendall A, Cipolla R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(12): 2481-2495.

[12] Bescos B, Fácil J M, Civera J, Neira J. DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes[J]. IEEE Robotics and Automation Letters, 2018, 3(4): 4076-4083.

[13] He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2961-2969.

[14] Shao C, Zhang L, Pan W. Faster R-CNN Learning-Based Semantic Filter for Geometry Estimation and Its Application in vSLAM Systems [J]. IEEE Transactions on Intelligent Transportation Systems, 2021.

[15] G. Singh, M. Wu, M. V. Do and S. -K. Lam, "Fast Semantic-Aware Motion State Detection for Visual SLAM in Dynamic Environment," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 12, pp. 23014-23030, Dec. 2022, doi: 10.1109/TITS.2022.3213694.

[16] Chen Y, Li N, Zhu D, Zhou C, Hu Z, Bai Y, Yan J. Bevsoc:Self-supervised contrastive learning for calibration-free bev 3d object detection. IEEE Internet of Things Journal (2024).

[17] Qi H, Fu Y, Hu Z, Wu J, Zhao Y. A lightweight semantic vslam approach based on adaptive thresholding and speed optimization. Journal of Beijing University of Aeronautics and Astronautics (2024).

[18] Li R, Zhao Y, Hu Z, Qi W,Liu G. Tohf: A feature extractor for resource -constrained indoor vslam. Journal of System Simulation, 1-12.

[19] Yang, Supeng, et al. "Visual SLAM Algorithm Based on YOLOv5 in Dynamic Scenario." *2023 China Automation Congress (CAC)*. IEEE, 2023.

[20] Wang C Y, Bochkovskiy A, Liao H Y M. YOLOv7:Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[J]. arXiv preprint arXiv:2207.02696, 2022.

[21] Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv:2004.10934, 2020.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TITS.2024.3520177, IEEE Transactions on Intelligent Transportation Systems

[22] Ai Y, Rui T, Lu M, Fu L, Liu S, Wang S. DDL-SLAM: A robust RGB-D SLAM in dynamic environments combined with deep learning[J]. IEEE Access, 2020, 8: 162335-162342.

[23] Fan Y, Zhang Q, Liu S, Tang Y, Jing X, Yao J, Han H. Semantic SLAM with more accurate point cloud map in dynamic environments[J]. IEEE Access, 2020, 8: 112237-112252.

[24] Han S, Xi Z. Dynamic scene semantics SLAM based on semantic segmentation[J]. IEEE Access, 2020, 8: 43563-43570.

[25] Chang Z, Wu H, Sun Y, Li C. RGB-D Visual SLAM Based on Yolov4-Tiny in Indoor Dynamic Environment. Micromachines. 2022; 13(2):230.

[26] Azimi, A., Ahmadabadian, A.H. and Remondino, F. PKS: A photogrammetric key-frame selection method for visual-inertial systems built on ORB-SLAM3. ISPRS Journal of Photogrammetry and Remote Sensing, 2022, 191, pp.18-32.

[27] Hosseininaveh A, Serpico M, Robson S, Hess M, Boehm J, Pridden I, Amati G. Automatic image selection in photogrammetric multi-view stereo methods [C]// Proc of Eurographics Association, 2012.

[28] Bampis L, Amanatiadis A, Gasteratos A. Fast loop-closure detection using visual-word-vectors from image sequences[J]. The International Journal of Robotics Research, 2018, 37(1): 62-82.

[29] Liu Z, Mao H, Wu C Y, Feichtenhofer C. A convnet for the 2020s[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 11976-11986.

[30] Z. Hu, W. Qi, K. Ding, G. Liu and Y. Zhao, "An Adaptive Lighting Indoor vSLAM With Limited On-Device Resources," in *IEEE Internet of Things Journal*, doi: 10.1109/JIOT.2024.3406816.

[31] Choi, Youn-Ho, and Seok-Cheol Kee. "Monocular Depth Estimation Using a Laplacian Image Pyramid with Local Planar Guidance Layers." *Sensors* 23.2 (2023): 845.

[32] Zhang H, Hu W, Wang X. Parc-net: Position aware circular convolution with merits from convnets and transformer[C]//Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI. Cham: Springer Nature Switzerland, 2022: 613-630.

[33] Woo S, Park J, Lee J Y, Kweon I. Cbam: Convolutional block attention module[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.

[34] Arandjelovic R, Gronat P, Torii A, Pajdla T, Sivic J. NetVLAD: CNN architecture for weakly supervised place recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 5297-5307.

[35] Liu Z, Lin Y, Cao Y, Hu H, Wei Y. Swin transformer: Hierarchical vision transformer using shifted windows[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 10012-10022.

[36] Jégou H, Douze M, Pérez P, Schmid C. Aggregating local descriptors into a compact image representation[C]//2010 IEEE computer society conference on computer vision and pattern recognition. IEEE, 2010: 3304-3311.

[37] 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, IEEE, 2012, pp. 573–580.

[38] Torii A, Sivic J, Pajdla T, Okutomi M. Visual place recognition with repetitive structures[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2013: 883-890.

[39] Torii A, Arandjelovic R, Sivic J, Okutomi M, Pajdla T. 24/7 place recognition by view synthesis[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1808-1817.

[40] Cui L, Ma C. SOF-SLAM: A semantic visual SLAM for dynamic environments[J]. IEEE access, 2019, 7: 166528-166539.

[41] Zhao X, Zuo T, Hu X. OFM-SLAM: a visual semantic SLAM for dynamic indoor environments[J]. Mathematical Problems in Engineering, 2021, 2021: 1-16.