RESEARCH ON YOLOv5-BASED VISUAL SLAM OPTIMISATION METHOD IN FARM DEPOT ENVIRONMENT
/
农场仓库环境中基于 YOLOv5 的视觉 SLAM 优化方法研究

Pengcheng LV [1,2], Zhenwei LI [2,*]
[1] Shandong University of Technology, College of Agricultural Engineering and Food Science, ZiBo, China
[2] Hezhou University, School of Artificial Intelligence, HeZhou, China
Tel: +86 18077508818; E-mail: lizhenwei0511@163.com
Corresponding author: Zhenwei LI
DOI: https://doi.org/10.35633/inmateh-75-07

Keywords: *Dynamic environments; YOLOv5; Synchronized positioning and mapping; Robustness*

ABSTRACT
Conventional simultaneous localization and mapping (SLAM) systems for agricultural robots rely heavily on static rigidity assumptions, which makes it susceptible to the influence of dynamic target feature points in the environment thus leading to poor localization accuracy and robustness of the system. To address the above issues, this paper proposes a method that utilizes a target detection algorithm to identify and eliminate dynamic target feature points in a farm depot. The method initially employs the YOLOv5 target detection algorithm to recognize dynamic targets in the captured warehouse environment images. The detected targets are then integrated into the feature extraction process at the front end of the visual SLAM. Next, dynamic feature points belonging to the dynamic target part are eliminated from the extracted image feature points using the LK optical flow method. Finally, the remaining feature points are used for location matching, map construction and localization. The final test on the TUM dataset shows that the enhanced vision SLAM system improves the localization accuracy by 91.47% compared to ORB-SLAM2 in highly dynamic scenes. This improvement increases the accuracy and robustness of the system and outperforms some of the best SLAM algorithms while maintaining high real-time performance. These features make it more valuable for mobile devices.

摘要
农业机器人的传统同步定位和地图构建（SLAM）系统在很大程度上依赖于静态刚性假设，这使得它很容易受到环境中动态目标特征点的影响从而导致系统的定位精度和鲁棒性变差。针对上述问题，本文提出了一种利用目标检测算法来识别和消除农场库房中动态目标特征点的方法。该方法最初采用 YOLOv5 目标检测算法来识别采集库房环境图像中的动态目标。然后将检测到的目标整合到视觉 SLAM 前端的特征提取过程中。接着，使用 LK 光流方法从提取的图像特征点中剔除属于动态目标部分的动态特征点。最后，剩余的特征点用于位置匹配、地图构建和定位。在 TUM 数据集上的最终测试表明，在高动态场景中，增强型视觉 SLAM 系统与 ORB-SLAM2 相比，定位精度提高了 91.47%。这一改进提高了系统的准确性和鲁棒性，并在保持高实时性的同时超越了一些优秀的 SLAM 算法。这些特点使其对移动设备更有价值。

INTRODUCTION

With the development of autonomous mobile robot platforms, agricultural robots have been widely used in agricultural production and warehousing services, such as farm management, orchard inspection, fruit picking, and automation of warehousing tasks. In these scenarios, robots need to understand the entire area and the precise location of the target objects in the map to accomplish autonomous navigation. In order to realize autonomous navigation, mobile robots need to accomplish two tasks: attitude estimation and map construction. Simultaneous localization and mapping (SLAM) refers to the robot in an unknown environment, which through its own matching sensors estimates its own position and build the environment map *(He et al., 2020)*. SLAM according to the different sensors are mainly divided into two categories. One is SLAM equipped with LiDAR, which is a mature system with small computation and accurate ranging, but the cost of LiDAR is high and not easy to maintain, so it is not commonly used in indoor robots. The other category is the camera-equipped vision SLAM, which is characterized by low cost, high cost-effectiveness, and the ability to obtain rich environmental information, so it has become a hot spot of attention in the field of robotics research.

Currently, visual SLAM can be classified into two kinds according to the methods used: the feature point method with FAST corner points as feature extraction and BRIEF descriptors as identity information matching, which can be used for sparse point cloud building; and the direct method with the information of the image gray value to directly judge the camera motion, which can be used for dense point cloud building, but with certain assumptions on the gray invariance. ORB-SLAM2 is considered to be one of the most complete one of the visual SLAM frameworks and also represents the feature point method *(Mur et al., 2017)*, but its results are not satisfactory in highly dynamic working environments, which leads to the low applicability of SLAM systems in real-world scenarios. The implementation of the direct method is based on the assumption of constant gray scale, but the light in the environment changes from time to time, and the assumption is difficult to be completely valid, so the SLAM system based on the direct method has poor robustness. In indoor dynamic environments, the feature points extracted from irregularly changing moving objects will seriously affect the accuracy of the camera position evaluation. *Engel et al., (2014)*, proposed LSD-SLAM, which utilizes gray values to achieve localization and construct semi-dense point cloud stacks. *Wang et al., (2017)*, contributed DSO-SLAM based on the sparse direct method is superior to LSD-SLAM in terms of robustness, accuracy, and speed, but it does not include the loopback detection function, which is an incomplete SLAM algorithm.

Among the approaches relying on deep learning, *Bescos et al., (2018)*, proposed the DynaSLAM algorithm, which utilizes a priori information for segmenting dynamic targets by means of a deep learning neural network, Mask R-CNN, which was first proposed by *He et al., (2017),* and *Liu et al., (2018),* utilized semantic segmentation to identify the a priori dynamic regions of an image, and tracking and mapping using static feature points. *Yu et al., (2018)*, proposed a DS-SLAM algorithm that combines a SegNet real-time semantic segmentation network with motion consistency detection to reduce the impact of dynamic targets on the system and reduce the localization accuracy in dynamic scenes compared to ORB-SLAM2. Compared with ORB-SLAM2, the localization accuracy in dynamic scenes is improved by one order of magnitude, but the semantic segmentation is time-consuming and fails to meet the real-time requirements. RDS-SLAM proposed by *Liu et al., (2021),* adds semantic tracking threads and optimization threads on the basis of the ORB-SLAM3 system and eliminates the outliers of the tracking threads by using the data correlation algorithm. The RTD-SLAM proposed by *Wang et al., (2023),* adds YOLOV5-based parallel semantic threads and optical flow modules to the tracking threads to eliminate dynamic feature points, which improves the system's localization accuracy and real-time performance. Based on the ORB-SLAM3 system, the semantic segmentation thread is added to improve the camera localization accuracy in dynamic scenes, but the dynamic feature points of potential dynamic targets (e.g., books held by people) are detected as static feature points, which are used in the tracking thread for feature matching and camera pose computation, resulting in a decrease in the system localization accuracy *(Law et al., 2018)*.

In summary, the SLAM system's positioning accuracy may be affected by dynamic objects in the complex environment, leading to poor real-time performance. To tackle these issues, this paper selects YOLOv5 as the target detection network. Then, it combines the optical flow method in the tracking thread of the SLAM system to eliminate feature points that do not meet the requirements. Finally, only the processed feature points are utilized for estimating the camera position. Experiments were conducted on the TUM dataset and compared with ORB-SLAM2 and other dynamic SLAM algorithms. The results showed that the localization accuracy of the improved visual SLAM system was 91.47% higher than that of ORB-SLAM2. Compared to ORB-SLAM2, the improved visual SLAM system demonstrated a 91.47% increase in localization accuracy in highly dynamic scenes. This enhancement effectively improves the system's robustness and localization accuracy, while also providing higher real-time performance on mobile devices. As a result, the system has greater application value.

MATERIALS AND METHODS

ORB-SLAM2 system

ORB-SLAM2 is a feature-point based SLAM system that enables simultaneous localization and map construction using camera-captured image data. It is highly stable, operates quickly, and is easy to implement. Currently, it is the most widely used system in the field of vision SLAM. The system contains three main threads: tracking, local map construction, and closed-loop detection. Figure 1 shows the system framework.

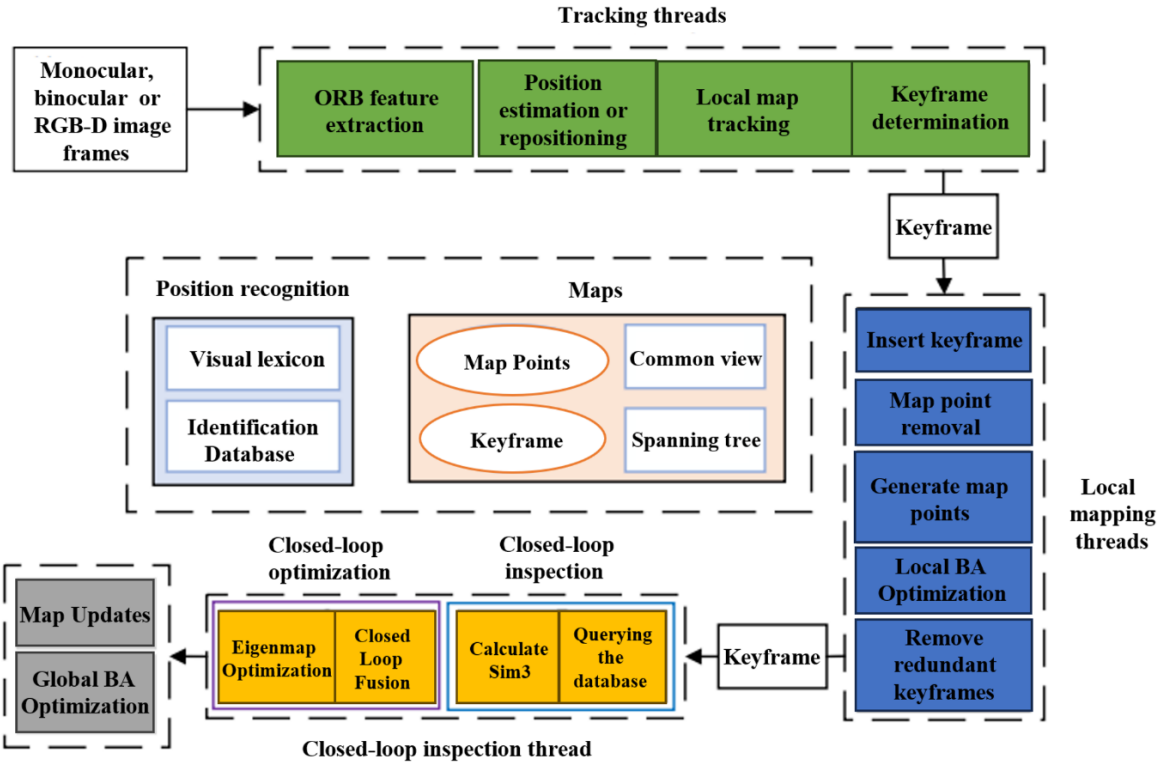


Fig. 1 – ORB-SLAM2 algorithm system framework

Improved SLAM system

The ORB feature points extracted from dynamic objects in the traditional tracking thread can accumulate errors when matching positions, resulting in reduced position evaluation accuracy or even localization failure *(Liang et al., 2022)*. To minimize the impact of dynamic objects on the system and improve positioning accuracy, this paper proposes a method that combines target detection and optical flow algorithms to reject dynamic feature points in the scene *(Liang et al., 2022)*. A target detection module and a dynamic feature point rejection module have been added to the tracking thread of the ORB-SLAM2 framework, as shown in Figure 2. A new detection thread has been included in the front-end, and the tracking and detection threads share information *(Placed et al., 2022)*. When the system receives the image frames, they are processed by the tracking and detection thread. The process involves two threads: tracking and detection. The tracking thread extracts ORB feature points from the image and uses the optical flow method to track and match the remaining feature points. The detection thread recognizes the object based on a priori information, such as the screen, chair, and human, and calculates the frame position of each category. The tracking thread then divides the frames into dynamic and static categories based on the returned frame information and categories, and calculates the basis matrix. The system's robustness and positioning accuracy are improved by matching the remaining static points with features to estimate their position, which reduces the influence of dynamic objects in the environment *(Khan et al., 2022; Wang et al., 2022; Tian et al., 2023; Engel et al., 2017)*.

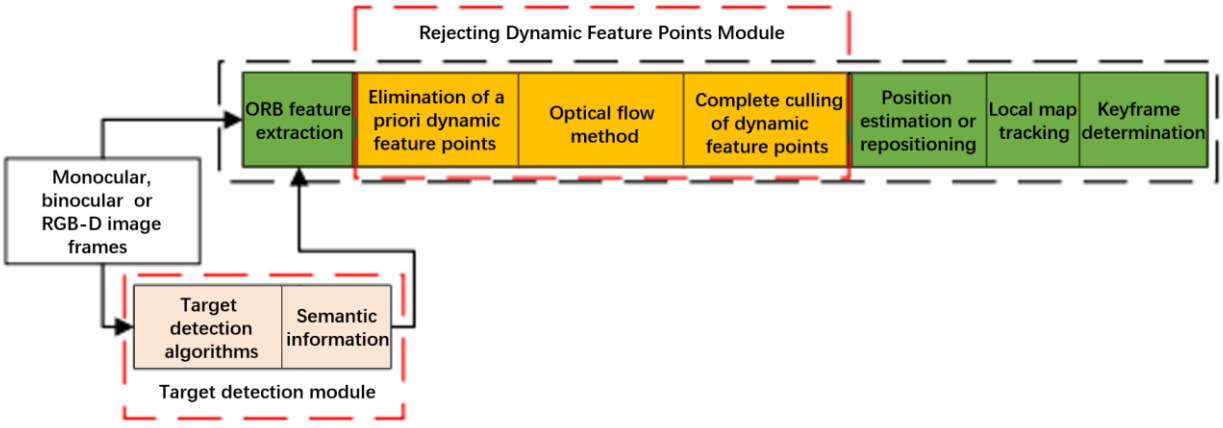


Fig. 2 – Improved trace threads

Feature tracking and matching based on the LK optical flow method

The optical flow method can be used for computer vision tasks by comparing the change in brightness values of corresponding pixels in two images to infer the direction and magnitude of the pixel's motion in the image. This method can track the motion process of a stationary pixel and a dynamic target's same pixel in the image *(Xu et al., 2021)*. The two types of optical flow are sparse optical flow, which describes the motion state of some pixels in the image, and dense optical flow, which describes the motion state of all pixels *(Shen et al., 2023)*. The Hom-Schunck optical flow represents the dense optical flow, while the Lucas-Kanade optical flow, also known as LK optical flow, dominates the sparse optical flow *(Zou et al., 2022)*.

In this paper, the purpose is to reduce computation by computing only the optical flow field of the ORB feature points extracted by the visual odometry of the SLAM system. Therefore, the LK optical flow is used *(Fang et al., 2009)*.

The LK optical flow is founded on 3 assumptions:

1. For a moving target in a grayscale image, the luminance (gray scale) of its pixel points does not change between adjacent frames.

2.Time continuity or motion is small enough that there is no drastic change in the target position due to time change in each computation, and the change in the corresponding position of pixel points of a moving target between adjacent frames is relatively small.

3. Spatial consistency, the vicinity of feature points All neighboring pixel points move similarly.

As shown in Fig. 3, the gray scale of an image can be regarded as a function of time: at moment t, the gray scale of an ORB feature point located at (x, y) in the image can be written as $I(x, y, t)$. According to the gray scale invariance assumption of the optical flow method, the gray scale value of the same feature point is fixed in each image. For a feature point located at (x, y) at time t, it will move to $(x+dx, y+dy)$ at time $(t+dt)$ *(Zhang et al., 2020)*.

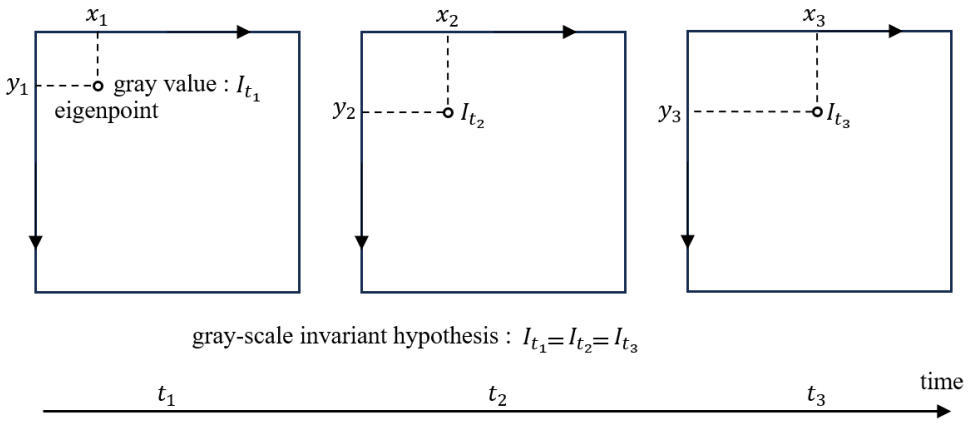


Fig. 3 – Schematic diagram of LK optical flow method

The following mathematical formula is obtained based on the assumption of gray scale invariance:

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

A Taylor expansion of the left-hand side of the equal sign of Eq. (1), retaining the first-order terms yields Eq. (2):

$$I(x+dx,\ y+dy,\ t+dt) \approx I(x,\ y,\ t)+\frac{\partial I}{\partial x}dx+\frac{\partial I}{\partial y}dy+\frac{\partial I}{\partial t}dt \tag{2}$$

According to the gray scale invariance assumption, the gray scale values of the feature points at moments t and $t+dt$ are equal, which can be obtained:

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

Both sides are obtained by dividing by dt at the same time:

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

In Eq. (4) dx/dt is the motion velocity of the feature point on the *X-axis* and dy/dt is the velocity on the *Y-axis*, which are denoted as u and v, respectively *(Kundu et al., 2009)*.

$\partial I/\partial x$ is the gradient of the image in the X-*axis* direction at the point, and $\partial I/\partial y$ is the gradient in the *Y-axis* direction, which are denoted as I_X and I_y, respectively *(Migliore et al., 2009)*. The amount of change in the grayscale of the feature point with respect to the time is denoted as it. It is written in the form of a matrix, as shown in Eq. (5) as follows *(Lin et al., 2010):*

$$\begin{bmatrix} I_x I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -I_t \tag{5}$$

According to Eq. (5), additional constraints need to be introduced to find the velocity vector of a pixel *(Zou et al., 2012)*. In the LK optical flow, a 6 × 6 window is assumed with the feature point as the center, and according to Assumption 3, the 36 pixels inside have the same motion, and the equation is transformed into a super-definite linear equation about u, v. The equation is transformed into a hyper-definite linear equation about, and solved by the least squares method *(Du et al., 2020)*. Since in the actual application scenario, the static part of the image generates optical flow vectors due to the movement of the camera, the average optical flow vectors of the static part of the image are calculated according to Equation (6) *(You et al., 2023)*.

$$\begin{bmatrix} U \\ V \end{bmatrix} = \frac{1}{N} \sum_{k=1}^{N} \begin{bmatrix} u_k \\ v_k \end{bmatrix} \tag{6}$$

The feature points are filtered using equation (6) to determine whether the feature points are dynamic or not *(Xiao et al., 2019)*.

$$\sqrt{(u-U)^2 + (v-V)^2} > z \tag{7}$$

where z is the threshold value for determining whether the feature point is a dynamic feature point, which is generally twice the static mean optical flow vector. If it is greater than this value, it is judged to be a dynamic feature point, and vice versa for a static feature point *(Zhong et al., 2018)*.

Dynamic property setting for indoor targets

The YOLO algorithm is a neural network-based object recognition and localization tool that is fast and can be used in real-time systems. It is currently one of the most widely used single-stage target detection algorithms. The YOLOv5 version contains five models: YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLO5x. The detection accuracy of the YOLOv5 model gradually increases from YOLOv5n to YOLO5x, while the detection speed gradually decreases. YOLOv5 employs mosaic data enhancement technology to enhance the generalization of the model by synthesizing multiple images into one large image. Secondly, it performs adaptive optimization in anchor point and image scaling processing to improve the performance of the model. For the backbone layer, YOLOv5 combines Focus and CSP structures to extract more representative features. The neck network introduces FPN and PAN structures to realize multi-scale feature fusion and improve detection accuracy. Finally, in the head output layer, the loss function GIOU_Loss and the predictive frame filtering GIOU_nms are improved to increase the accuracy and recall of the model. *(Wu et al., 2021; Redmon et al., 2016; Redmon et al., 2017; Redmon et al., 2018)*.

This paper selects the widely used YOLOv5s network for dynamic target detection due to its better balance between accuracy and speed. YOLOv5s is only 27MiB in size and has a fast inference speed, meeting the real-time detection requirements of the visual system when compared to YOLOv4 *(Liu et al., 2016; Lin et al., 2017; Bochkovskiy et al., 2020)*.

The data structure of the YOLOv5 detection frame is output in the format (X, Y, W, H, class, confidence), where X and Y represent the X and Y coordinates of the center point of the detection frame, respectively, W and H represent the width and height of the frame, class represents the category, and confidence represents the confidence level. In order to facilitate reading in the SLAM system, it is necessary to transform the first 4 position information into the coordinates under the original image. The conversion formula is as follows:

$$\begin{cases} X_1 = \left(X - \frac{W}{2} \right) \times L \\ X_2 = \left(X + \frac{W}{2} \right) \times L \\ Y_1 = \left(Y + \frac{H}{2} \right) \times D \\ Y_2 = \left(Y - \frac{H}{2} \right) \times D \end{cases} \quad (8)$$

The current approach defines dynamic and static boxes based on given coordinates and image dimensions. Where (X_1, Y_1) and (X_2, Y_2) denote the upper left and lower right coordinates of the detection frame, and *L* and *D* denote the width and height of the image respectively. The detection frame for the human body is defined as dynamic, while the detection frames for other objects are tentatively defined as static. Once the ORB feature points are extracted, the tracking thread receives the detection data from YOLOv5 and traverses the feature points in the detection frame. The feature points are categorized according to the definition of different frames. Figure 4 depicts the schematic diagram of feature points. To determine whether a feature point is dynamic or static when two boxes overlap, check if it is located inside the dynamic box and outside the static box. If the condition is met, the feature point is considered dynamic. If not, it is considered static *(Girshick et al., 2014; Borrego et al., 2018; Ren et al, 2015)*.

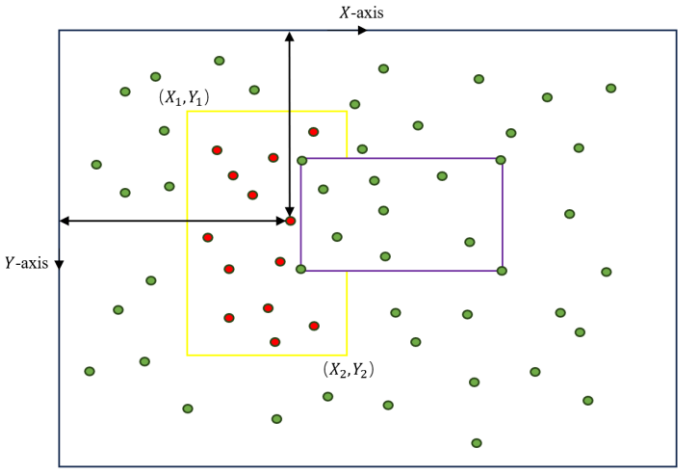


Fig. 4 – Schematic diagram of feature points

The current strategy for rejecting dynamic feature points necessitates the dynamic detection of all feature points within the target detection anchor frame. While this approach can improve localization accuracy in low dynamic sequences, it negatively impacts the system's real-time performance. To address this issue, this paper proposes a new dynamic feature point rejection strategy. This strategy employs a priori knowledge to classify targets. For instance, frequently moving targets like people, animals, and sweeping robots are classified as dynamic targets, while targets like tables and chairs that may move due to human movement are classified as potential dynamic targets.

Table 1

Dynamic property setting for indoor targets

Target objects	Target category	Target objects	Target category
Human	a	Desk	b
Dog	a	Chair	b
Cat	a	Water cup	b
Bird	a	Book	b
Pig	a	Laptop	b

Table 1 displays the categorization of common indoor targets, where 'a' denotes a dynamic target and 'b' denotes a potential dynamic target. Based on these classifications, feature points located in the anchor frame of dynamic targets are classified as dynamic feature points, feature points located in the anchor frame of potentially dynamic targets are classified as potentially dynamic feature points, and the remaining feature

points are classified as static feature points. Assuming that the set of dynamic feature points in an image frame is Z, the set of potential dynamic feature points is Q, and the set of static feature points is P, the number of feature points that need to be dynamically detected can be reduced. This improves the real-time performance of the system.

Using Fig. 5 as an example, the tracking thread of the SLAM system extracts feature points while employing the YOLOv5s target detection network for target detection. Feature points in the anchor frames of dynamic targets (e.g. people in the two figures) are placed in set Z. Feature points in the anchor frames of potentially dynamic targets (e.g. the computer screen and chair in the figure) are placed in set Q, and the remaining feature points are placed in set P.

Calculate the optical flow vectors of the feature points in sets Q and P using the Lucas-Kanade optical flow method. The optical flow vectors of the feature points in set P are used as the optical flow vectors of the static region and are substituted into Equation (6) to calculate the average optical flow vectors. The optical flow vectors of the feature points in set Q are substituted into Equation (7) to make a judgment. If they are less than the threshold value, it means that the feature point is a static feature point and is then moved into set P of the static feature points. Finally, the feature points in set P are retained for feature matching and camera pose estimation, and the remaining feature points are rejected.

(a) Before removing feature points (b) After feature removal

(c) Before removing feature points (d) After feature removal

Fig. 5 – Dynamic feature point rejection effect comparison

Our strategy utilizes the LK optical flow method to determine whether a feature point is affected by a dynamic target, rather than simply removing all feature points in the target detection frame. An optical flow vector threshold is set; if the optical flow vector is less than this threshold, it indicates that the feature point is not affected by the dynamic target and can be added to the set P. Increasing the number of feature point matches can further improve the localization accuracy and robustness of the system. The effect of dynamic feature point rejection is demonstrated in Fig. 5. The algorithm rejects feature points on the person as dynamic and retains feature points on the computer anchor frame that overlap with the person's anchor frame, provided that the optical flow vector is less than the threshold. Feature points on the chair's anchor frame portion are also rejected as dynamic due to the person's movement. This strategy enables more accurate rejection of dynamic feature points, resulting in improved system stability and accuracy.

RESULTS

Experiment details

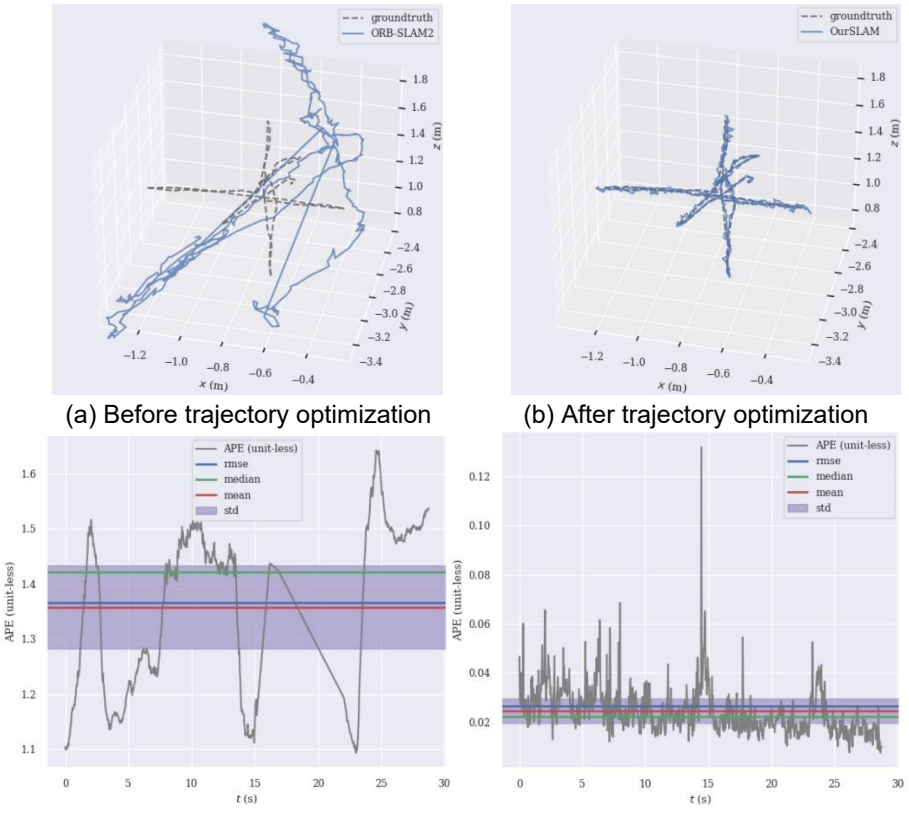
To evaluate the effectiveness of the improved ORB-SLAM2 algorithm, experiments and tests using the TUM dataset were conducted. This dataset consisted of synchronized RGB images and depth images of an indoor warehouse scene captured by a robot equipped with a Kinect sensor. It became one of the most widely used evaluation datasets in the field of SLAM for comparing the performance of different algorithms. This paper tested six sequences from the TUM dataset, including sitting_xyz, sitting_static, walking_halfsphere, walking_rpy, walking_static, and walking_xyz. The sitting_xx sequence was a low-dynamic scenario, while the walking_xx sequence was a high-dynamic scenario. Generally, SLAM algorithm evaluation considered aspects such as time consumption, complexity, and accuracy.

Accuracy evaluation was often the most important, and it involved metrics such as absolute and relative trajectory errors. This paper used root mean square error (RMSE) and standard deviation (STD) to evaluate these metrics. To reduce the impact of tracking failures in dynamic sequences during experiments, each sequence was run fifty times and the average value was recorded as the experimental data.

The experimental equipment used for the experiment was a Shenzhou laptop, its CPU model was I7-12650H, memory was 16G, the graphics chip was NVIDIA Geforce GTX4060, the system environment was Ubuntu20.04, and the deep learning framework PyTorch 1.9.0 was loaded in the virtual environment of Anaconda. The target detection experiment running software was Visual Studio Code. The target detection network was written in Python3.6, and the SLAM part was written in C++.

Analysis of experimental results

Since the system in this paper is an improvement on the ORB-SLAM2 system, the improved system with the ORB-SLAM2 system is compared and the evo tool is used to compare the bitwise trajectories estimated by the algorithm in this paper and the ORB-SLAM2 algorithm with the real trajectory map groundtruth.txt given by the dataset to quantify the effect of the algorithm in this paper on the improvement effect of the SLAM algorithm.



(a) Before trajectory optimization (b) After trajectory optimization

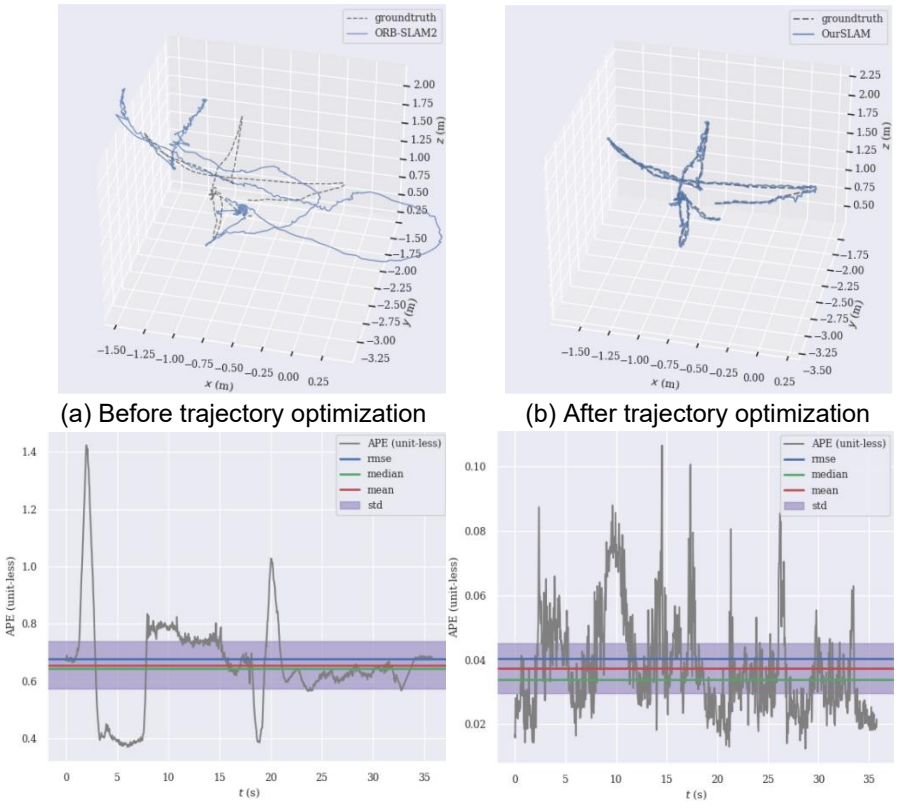
(c) Before ATE optimization (d) After ATE optimization

Fig. 6 – Comparison of trajectories and errors of datasets on walking_xyz

Figures 6 and 7 compare and analyze the camera trajectories evaluated by ORB-SLAM2 and this algorithm on the walking_xyz and walking_halfsphere datasets, respectively. The dotted line represents the

real trajectory, while the solid line represents the camera trajectory evaluated by ORB-SLAM2 and this algorithm. The absolute trajectory error is denoted as (ATE). Figures 6 and 7 demonstrate the algorithm's accuracy by showing the similarity between its trajectories and the real trajectories. Additionally, the improved algorithm (Figures 6(d) and 7(d)) significantly reduces various types of error values compared to the unimproved algorithm.

To verify the effectiveness of the experimental design, ablation experiments were conducted on the TUM dataset. ORB-SLAM2 served as the base group, and the base group + YOLOv5s (ORB+YOLO) and the base group + LK optical flow method (ORB+LK), as well as the improved algorithm proposed in this paper, were tested.



(a) Before trajectory optimization (b) After trajectory optimization

(c) Before ATE optimization (d) After ATE optimization

Fig. 7 – Comparison of trajectories and errors of datasets on walking_halfsphere

The comparison results are presented in Table 2, which demonstrate the superior effectiveness of the improved algorithm.

Table 2

Comparison results of ablation experiments

Data set	ORB-SLAM2		ORB + YOLO		ORB + LK		Ours	
	Rmse	STD	Rmse	STD	Rmse	STD	Rmse	STD
walking_xyz	0.2991	0.1186	0.0197	0.0116	0.1894	0.0752	0.0185	0.0092
walking_halfsphere	0.4754	0.1676	0.0411	0.0327	0.0752	0.0436	0.0381	0.0161
walking_static	0.0925	0.0714	0.0098	0.0087	0.0168	0.0114	0.0094	0.0082
walking_rpy	0.3575	0.2364	0.0468	0.0359	0.0748	0.0367	0.0349	0.0249
sitting_xyz	0.0153	0.0068	0.0161	0.0052	0.0151	0.0067	0.0134	0.0052
sitting_static	0.0122	0.0043	0.0124	0.0051	0.0097	0.0048	0.0082	0.0030

In this paper, the algorithm is simulated on six datasets in high and low dynamic environments, and the improvement degree of the improved algorithm over ORB-SLAM2 is calculated as shown in Eq. (9), which visually expresses the optimization effect.

$$\rho = \frac{\mu - \gamma}{\mu} \times 100\% \tag{9}$$

In the formula: ρ is the degree of improvement; μ is the result data of ORB-SLAM2 algorithm; γ is the result data of this paper's algorithm.

Table 3 and Table 4 show that the improved algorithm has an average improvement rate of RMSE and STD of less than 25% for the two sitting datasets in low dynamic scenarios, which is suboptimal. In datasets with dynamic objects, the removal of portrait feature points has little effect on the system's normal operation in low dynamic scenarios. However, in high dynamic scenarios, the algorithm presented in this paper shows an average improvement of 91.47% in the RMSE of four dynamic datasets compared to the ORB-SLAM2 algorithm. This indicates that the algorithm in this paper provides better localization accuracy in high dynamic scenarios.

Table 3

Comparison of absolute trajectory errors between ORB-SLAM2 and our algorithm

Data set	ORB-SLAM2		Ours		Relative uplift rate/%	
	Rmse	STD	Rmse	STD	Rmse	STD
walking_xyz	0.2991	0.1186	0.0185	0.0092	93.815	92.243
walking_halfsphere	0.4754	0.1676	0.0381	0.0161	91.986	90.394
walking_static	0.0925	0.0714	0.0094	0.0082	89.838	88.515
walking_rpy	0.3575	0.2364	0.0349	0.0249	90.238	89.467
sitting_xyz	0.0153	0.0068	0.0134	0.0052	12.418	23.529
sitting_static	0.0122	0.0043	0.0082	0.0030	32.787	30.233

In recent years, scholars have presented numerous cases using the fusion of deep learning and optical flow methods. This paper compares the reliability of the algorithm with recent domestic and international dynamic vision SLAM algorithms, as shown in Table 5. The compared algorithms include DynaSLAM, DS-SLAM, and RDS-SLAM, with the root-mean-square error of the absolute trajectory path as the comparative data. The comparison data is the root mean square error of the absolute trajectory path. To eliminate the influence of other factors, such as hardware equipment, this paper calculates the relative improvement rate. This directly illustrates the improvement rate of the enhanced algorithm compared to the original SLAM algorithm under the same experimental conditions. The DynaSLAM algorithm and the algorithm proposed in this paper have shown the best results. However, the DynaSLAM algorithm's use of a semantic segmentation algorithm for dynamic feature point rejection consumes a lot of time, making it unsuitable for real-time requirements. In contrast, this paper's algorithm has demonstrated relatively impressive localization accuracy in highly dynamic scenes, confirming its reliability.

Table 4

Comparison of relative trajectory errors between ORB-SLAM2 and our algorithm

Data set	ORB-SLAM2		Ours		Relative uplift rate/%	
	Rmse	STD	Rmse	STD	Rmse	STD
walking_xyz	0.2107	0.1079	0.0163	0.0088	92.264	91.844
walking_halfsphere	0.3247	0.1642	0.0274	0.0142	91.561	91.352
walking_static	0.0372	0.0612	0.0042	0.0063	88.710	89.706
walking_rpy	0.3547	0.4665	0.0351	0.0462	90.104	90.096
sitting_xyz	0.0167	0.0062	0.0162	0.0061	2.994	1.613
sitting_static	0.0114	0.0018	0.0111	0.0017	2.632	5.556

Table 5

Comparison of ATE analysis between the improved algorithm and other dynamic SLAM algorithms

Data set	DynaSLAM	RDS-SLAM	DS-SLAM	Ours
walking_xyz	92.74/%	91.27/%	90.56/%	**93.82/%**
walking_halfsphere	**94.28/%**	90.74/%	89.38/%	91.99/%
walking_static	87.78/%	86.54/%	83.34/%	**89.84/%**
walking_rpy	**92.67/%**	90.13/%	86.52/%	90.24/%

CONCLUSIONS

This paper proposes a method that uses a target detection algorithm to identify and exclude feature points of dynamic targets in a farm depot. The method first utilizes the YOLOv5 target detection algorithm to

identify dynamic targets in the acquired environment images. The recognized targets are then integrated into the visual SLAM front-end for feature extraction. Next, an LK optical flow method is used to eliminate dynamic feature points that belong to the dynamic target portion of the extracted image feature points. The remaining feature points are then utilized for bit matching and map construction to determine the location of the feature points. Tests were also conducted on the TUM dataset to evaluate the performance of the proposed method. The experimental results show that the enhanced visual SLAM system improves the localization accuracy by 91%. In highly dynamic scenes, the system effectively improves the localization accuracy and robustness by 47% over ORB-SLAM2. In addition, compared with other excellent SLAM algorithms, the system has significantly improved localization accuracy and higher real-time performance, so it is more suitable for the application of mobile devices on agricultural robots, which will promote the development of smart agriculture.

ACKNOWLEDGEMENT

The project supported by the R&D of low-speed unmanned autonomous navigation controller Project under Grant (No.2022TSGC1175).

REFERENCES

[1] Bescos B, Fácil JM, Civera J. (2018). DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4):4076-4083. https://doi.org/10.1109/LRA.2018.2860039

[2] Bochkovskiy A, Wang CY, Liao HYM, (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:*200410934, . https://doi.org/10.48550/arXiv.2004.10934

[3] Borrego J, Figueiredo R, Dehban A. (2018). A generic visual perception domain randomization framework for gazebo. *2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC),* p.237-242. https://doi.org/10.1109/ICARSC.2018.8374189

[4] Du ZJ, Huang SS, Mu TJ. (2020). Accurate dynamic slam using CRF-based long-term consistency. *IEEE Transactions on Visualization and Computer Graphics*, 28(4):1745-1757. https://doi.org/10.1109/TVCG.2020.3028218

[5] Engel J, Koltun V, Cremers D. (2017). Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611-625. https://doi.org/10.1109/TPAMI.2017.2658577

[6] Engel J, Schöps T, Cremers D. (2014). LSD-SLAM: Large-scale direct monocular slam. *European conference on computer vision*, p.834-849. https://doi.org/10.1007/978-3-319-10605-2_54

[7] Fang Y, Dai B. (2009). An improved moving target detecting and tracking based on optical flow technique and Kalman filter. *2009 4th International Conference on Computer Science & Education,* p.1197-1202. https://doi.org/10.1109/ICCSE.2009.5228464

[8] Girshick R, Donahue J, Darrell T. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the *IEEE conference on computer vision and pattern recognition*, p. 580-587. https://doi.org/10.48550/arXiv.1311.2524

[9] He G, Yuan X, Zhuang Y. (2020). An integrated GNSS/LiDAR-SLAM pose estimation framework for large-scale map building in partially GNSS-denied environments. *IEEE Transactions on Instrumentation and Measurement*, 70:1-9. https://doi.org/10.1109/TIM.2020.3024405

[10] He K, Gkioxari G, Dollár P. (2017). Mask R-CNN. Proceedings of the *IEEE international conference on computer vision*, p.2961-2969. https://doi.org/10.48550/arXiv.1703.06870

[11] Khan MU, Zaidi SAA, Ishtiaq A. (2021). A comparative survey of lidar-slam and lidar based sensor technologies.*2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC)*, p.1-8. https://doi.org/10.1109/MAJICC53071.2021.9526266

[12] Kundu A, Krishna KM, Sivaswamy J, 2009. Moving object detection by multi-view geometric techniques from a single camera mounted robot. *(2009) IEEE/RSJ Inter-national Conference on Intelligent Robots and Systems*,p.4306-4312. https://doi.org/10.1109/IROS.2009.5354227

[13] Law H, Deng J. (2018). CornerNet: Detecting objects as paired keypoints. Proceedings of the *European conference on computer vision (ECCV)*, p.734-750. https://doi.org/10.1007/s11263-019-01204-1

[14] Liang T, Bao H, Pan W. (2022). Traffic sign detection via improved sparse R-CNN for autonomous vehicles. *Journal of Advanced Transportation*, 2022:1-16. https://doi.org/10.1155/2022/3825532

[15] Liang T, Bao H, Pan W. (2022). DetectFormer: category-assisted transformer for traffic scene object detection. *Sensors*, 22(13):4833. https://doi.org/10.3390/s22134833

[16] Lin KH, Wang CC. (2010). Stereo-based simultaneous localization, mapping and moving object tracking. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, p.3975-3980. https://doi.org/10.1109/IROS.2010.5649653

[17] Lin TY, Goyal P, Girshick R. (2017). Focal loss for dense object detection. Proceedings of the *IEEE international conference on computer vision,* p.2980-2988. https://doi.org/10.48550/arXiv.1708.02002

[18] Liu W, Anguelov D, Erhan D. (2016). SSD: Single shot multibox detector. *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, p.21-37*. https://doi.org/10.1007/978-3-319-46448-0_2

[19] Liu Y, Miura J. (2021). RDS-SLAM: Real-time dynamic slam using semantic segmentation methods. *IEEE Access*,9:23772-23785. https://doi.org/10.1109/ACCESS.2021.3050617

[20] Migliore D, Rigamonti R, Marzorati D. (2009). Use a single camera for simultaneous localization and mapping with mobile object tracking in dynamic environments. *ICRA Workshop on Safe navigation in open and dynamic environments: Application to autonomous vehicles*, p.12-17.

[21] Mur-Artal R, Tardós JD. (2017). ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE transactions on robotics*, 33(5):1255-1262. https://doi.org/10.1109/TRO.2017.2705103

[22] Placed JA, Strader J, Carrillo H. (2023). A survey on active simultaneous localization and mapping: State of the art and new frontiers. *IEEE Transactions on Robotics*, . https://doi.org/10.1109/TRO.2023.3248510

[23] Redmon J, Divvala S, Girshick R. (2016). You only look once: Unified, real-time object detection. Proceedings of the *IEEE conference on computer vision and pattern recognition*, p.779-788. https://doi.org/10.48550/arXiv.1506.02640

[24] Redmon J, Farhadi A. (2017). Yolo9000: better, faster, stronger. Proceedings of the *IEEE conference on computer vision and pattern recognition,* p.7263-7271. https://doi.org/10.48550/arXiv.1612.08242

[25] Redmon J, Farhadi A, 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv*:180402767, . https://doi.org/10.48550/arXiv.1804.02767

[26] Ren S, He K, Girshick R. (2015). Faster R-CNN: To-wards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28. https://doi.org/10.1109/TPAMI.2016.2577031

[27] Shen S, Kerofsky L, Yogamani S. (2023). Optical flow for autonomous driving: Applications, challenges and improvements. *arXiv preprint arXiv:230104422*, . https://doi.org/10.48550/arXiv.2301.04422

[28] Tian C, Liu H, Liu Z. (2023). Research on multi-sensor fusion SLAM algorithm based on improved Gmapping. *IEEE Access*, 11:13690-13703. https://doi.org/10.1109/ACCESS.2023.3243633

[29] Wang K, Yao X, Ma N. (2023). Real-time motion removal based on point correlations for RGB-D SLAM in indoor dynamic environments. *Neural Computing and Applications*, 35(12):8707-8722. https://doi.org/10.1007/s00521-022-07879-x

[30] Wang R, Schworer M, Cremers D. (2017). Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras. Proceedings of the *IEEE International Conference on Computer Vision*, p.3903-3911. https://doi.org/10.48550/arXiv.1708.07878

[31] Wang S, Huang Y, Yue P. (2022). Research progress on visual slam for dynamic environments. *International Workshop of Advanced Manufacturing and Automation*, p.108-115. https://doi.org/10.1007/978-981-19-9338-1_14

[32] Wu W, Liu H, Li L. (2021). Application of local fully convolutional neural network combined with YOLO v5 algorithm in small target detection of remote sensing image. *PLOS One*, 16(10):e0259283. https://doi.org/10.1371/journal.pone.0259283

[33] Xiao L, Wang J, Qiu X. (2019). Dynamic-slam: Semantic monocular visual localization and mapping based on deep learning in dynamic environment. *Robotics and Autonomous Systems*, 117:1-16. https://doi.org/10.1016/j.robot.2019.03.012

[34] Xu Z, Rong Z, Wu Y. (2021). A survey: which features are required for dynamic visual simultaneous localization and mapping? *Visual Computing for Industry, Biomedicine, and Art*, 4(1):1-16. https://doi.org/10.1186/s42492-021-00086-w

[35] You M, Luo C, Zhou H. (2023). Dynamic dense CRF inference for video segmentation and semantic slam. *Pattern Recognition*, 133:109023. https://doi.org/10.1016/j.patcog.2022.109023

[36] Yu C, Liu Z, Liu XJ. (2018). DS-SLAM: A semantic visual slam towards dynamic environments. *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, p.1168-1174. https://doi.org/10.1109/IROS.2018.8593691

[37] Zhang T, Zhang H, Li Y. (2020). FlowFusion: Dynamic dense RGB-D SLAM based on optical flow. 2020 *IEEE International Conference on Robotics and Automation (ICRA)*, p.7322-7328. https://doi.org/10.1109/ICRA40945.2020.9197349

[38] Zhong F, Wang S, Zhang Z. (2018). Detect-SLAM: Making object detection and SLAM mutually beneficial. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, p.1001-1010. https://doi.org/10.1109/WACV.2018.00115

[39] Zou L, Huang Z, Yu X. (2022). Automatic detection of congestive heart failure based on multiscale residual UNet++: From centralized learning to federated learning. *IEEE Transactions on Instrumentation and Measurement*, 72:1-13. https://doi.org/10.1109/TIM.2022.3227955

[40] Zou D, Tan P, (2012). CoSLAM: Collaborative visual SLAM in dynamic environments. *IEEE transactions on pattern analysis and machine intelligence*, 35(2):354-366. https://doi.org/10.1109/TPAMI.2012.104