Research Article

3D Semantic VSLAM of Dynamic Environment Based on YOLACT

Dongyan Wu,1 Bingbo Xie,2 and Chongben Tao2,3

1 Suzhou Vocational Institute of Industrial Technology, Suzhou, China
2 Suzhou University of Science and Technology, Suzhou, China
3 Tsinghua University Suzhou Automobile Research Institute, Suzhou, China

Correspondence should be addressed to Chongben Tao; tom1tao@163.com

Received 15 March 2022; Revised 12 July 2022; Accepted 27 July 2022; Published 7 September 2022

Academic Editor: Rahib Abiyev

Copyright © 2022 Dongyan Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A Visual Simultaneous Localization and Mapping (VSLAM) method is proposed to construct a 3D environment map in the creation of dynamic noise information which leads to significant errors in camera pose estimation and a substantial number of noise points. For this problem, this paper proposed a method based on YOLACT, which combined optical flow and ViBe+ semantic map construction algorithm. First, our approach uses LK optical flow method to estimate the overall motion trajectory of adjacent frames. Then the trajectory data are employed to intercept the relative position of the current frame to the previous frame. Afterward, we combine ViBe+ algorithm to accurately detect and eliminate dynamic noise. Secondly, image semantic segmentation is performed based on YOLACT model. Image feature points are extracted from MAPLAB algorithm, pose estimation of the camera is performed and movement trajectory is recorded to complete a semantic map. Finally, through ablations study with common algorithms, the experimental results show that the proposed algorithm effectively avoids interference of dynamic noise information on VSLAM. Additionally, the constructed semantic map provides higher precision, fewer noise points, and pretty robustness.

1. Introduction

Simultaneous Localization and Mapping (SLAM) technology has made great contributions to both driving applications and engineering applications, which has played a vital role in the emergence and development of autonomous driving [1]. The robot only uses cameras as a sensor to collect, filter, and calculate images of the surrounding environment. Completing the process of location determination and environment map construction is called Visual Simultaneous Localization and Mapping (VSLAM). Compared with other SLAM algorithms, VSLAM algorithms can additionally provide necessary environmental information about autonomous vehicles to help vehicle drivers determine important information such as seasons, weather, and traffic conditions [2]. With the development of artificial intelligence, intelligent robots are widely used in industrial productions, public services, and disaster relief. For example, contactless intelligent food delivery robots and autonomous navigation disinfection robots have been applied to new coronavirus epidemic. Whether it is engineering applications, autonomous driving, or intelligent robots, precise and stable autonomous navigation is the prerequisite for their work, so the study of SLAM technology has very important significance. VSLAM has gradually become the mainstream trend of SLAM due to its low cost, wide application scenarios, small device size, and ability to provide semantic information. In recent years, researchers have done a lot of research on VSLAM. Mur-Artal et al. [3] proposed ORB-SLAM2, which can realize map reconstruction, loop detection, relocation, improve the precision, and usability of VSLAM. Bloesch et al. [4] proposed CodeSLAM which derives a compact representation of a map by adjusting the gray map automatic encoder, which can optimize map reconstruction. Christian et al. [5] proposed Dense VSLAM which achieves the smallest pixel grayscale and the smallest depth error, obtains higher precision pose accuracy, and significantly reduces trajectory errors. Li et al. [6] made full use of structural features in the Manhattan world and proposed a monocular SLAM system, which can provide...
accurate camera poses and improve the precision of 3D maps. McCracken et al. [7] used Convolutional Neural Network (CNN) models and Bayesian update algorithm to track the probability distribution of each surface classification, then used the data provided by SLAM system to update these probabilities to improve semantic prediction. It is worth mentioning that NeuroSLAM proposed by Yu et al. [8] is a 3D environment-oriented brain-like SLAM system and a new set of brain-like SLAM system is developed for robots based on 3D navigation neural mechanism of brain.

Even though VSLAM technology has made great progress after several years of development, it is worth noticing that almost all theories and implementation schemes of the current mainstream VSLAM are based on static world assumptions, which requires objects of the environment must remain stationary during the VSLAM process. When the VSLAM environment is changed from an ideal static environment to a complex dynamic environment, a large amount of dynamic noise in the environment will have a serious impact on camera pose estimation during the VSLAM process. And the noise information retained in the point cloud map will have a serious impact on loop detection and semantic segmentation. Some scholars have conducted research on the low precision of VSLAM in a dynamic environment and have given some feasible solutions. These solutions can be divided into three categories, improve conventional dynamic detection algorithms, upgrade hardware equipment and introduce deep learning models. The static weighting methods of keyframe edge points [9], optimization of professional probability models [10], methods of combining Grid-based Motion Statistics (GMS) and sliding window adopted by DMS-SLAM [11], and estimation of motion residual methods [12] are all improvements that have been made based on conventional dynamic detection methods. Rebecq et al. [13] introduced a new hardware device event camera, making full use of the nature of the event camera and proposed a 3D environment map solution that is not affected by motion blur. Meanwhile, this method restores a semi-dense environment, which can operate well in highly 2 dynamic environments is robust to changes in light intensity. With the improvement of computer computing power, deep learning technology has become mature, providing a new direction for dynamic detection. Some scholars have introduced deep learning models to improve the precision of dynamic detection. Chao et al. [14] proposed DS-SLAM algorithm which combines a large number of segmentation clues to construct a Conditional Random Field (CRF) model, using particle filters to track motion to enhance motion detection capabilities, applying Maximum a Posteriori (MAP) estimation to vector depth images to accurately determine the foreground. Reddy et al. [15] proposed a model that introduces semantic constraint information on the reconstruction algorithm, which can significantly reduce relative error in the reconstruction of moving object trajectory. Esparza et al. [16] used a convolutional neural network, optical flow, and depth map to detect objects in the scene and process dynamic information in the environment. The proposed system has a fast processing time and can run in real time in outdoor and indoor environments. Saputra et al. [17] used Bayesian algorithms to track moving objects, enhancing the ability to detect, track, and reconstruct the shape of dynamic information to provide robust localization. It is worth mentioning that the paper introduces deep learning algorithms in this paper. Bescos et al. [18] proposed a dynamic object segmentation system DynaSLAM based on Mask RCNN and multi-view geometry, which increased the capabilities of dynamic object detection and background repair.

Although the above methods reduce the adverse impact of mobile on VSLAM, the following problems still exist. Such as DS-SLAM can only classify people as mobile categories and identify mobile robots or human-carried objects as static information, which greatly reduces the precision of VSLAM. Considering that image sequence processed by VSLAM algorithm contains multiple types of objects, there are still challenges in how to combine semantic information provided by deep learning with coordinate information provided by the VSLAM algorithm and apply it to construct 3D VSLAM semantic maps in indoor dynamic environments.

Above considerations mentioned before, this paper proposes a method based on YOLACT [19], which combines optical flow and ViBe+ [20] semantic map construction algorithm shown in the framework diagram in Figure 1. Compared with existing results, the semantic map construction algorithm proposed in this paper can filter out dynamic information, significantly reducing the impact of noise points trajectory on robot pose estimation and semantic segmentation in a 3D map. Meanwhile, the YOLACT model can improve the speed and precision of semantic segmentation. The high-precision semantic map constructed in this paper is more practical. Compared with other semantic mapping algorithms, the innovations of this paper are divided into the following two points. On the one hand, this paper proposes a dynamic rejection algorithm combining the optical flow method and ViBe+ while introducing YOLACT model in the dynamic rejection algorithm for further optimization. On the other hand, this paper also proposes a semantic database update algorithm to solve the problem of local semantic information mutation during the second scan.

The first section introduces how this paper combines the optical flow method with ViBe+ and realizes the elimination of dynamic information. In the second section, we introduce the YOLACT deep learning model used in this paper. The third section introduces how to use the depth camera to obtain data and combine the above algorithm to realize the construction of 3D semantic map in a dynamic environment. The fourth section shows and analyzes the experimental data. The fifth section is the conclusion of this paper.

2. Dynamic Information Detection

According to the spatial state of the camera, the visual dynamic detection methods are divided into two categories: one is based on a static camera and the other is based on a moving camera. When a camera is moving, image interference caused by the movement of the camera itself will
have a great impact on the first algorithm. In this case, the applicability of the first algorithm is not high. However, dynamic detection algorithms based on mobile cameras generally suffer from low detection precision. This paper solves the above-mentioned problems by combining the Lucas–Kanade (LK) optical flow and the ViBe+ method. The method is divided into three stages. First, LK optical flow method is used to estimate the overall motion trajectory of adjacent frames. Second, this trajectory data are used to intercept the current frame at the corresponding position of the previous frame. Finally, ViBe+ algorithm is used to accurately detect the moving object. And use the YOLACT model directly detects common absolutely moving objects, most of which are living objects, such as people, cats, and dogs. The algorithm combination frame diagram is shown in Figure 2.

LK optical flow method is a commonly used sparse optical flow method. The pyramid feature tracking algorithm is improved based on this method. First, a pyramid is built to calculate optical flow of the highest layer image, transfer the optical flow results obtained from the previous layer \((L + 1)\) to the next layer \((L)\), and calculate the residual light of this layer based on the \(L\) layer flow vector \(dL\). Since the size of each layer in the pyramid is half the size of the upper layer, the optical flow of each layer is half of the \(L\) layer optical flow, which not only ensures the residual optical flow vector of each layer is small but also can further apply the LK optical flow algorithm. Finally, the optical flow is iteratively calculated for each layer, and the resulting optical flow is also a superposition of all layers.

The initial optical flow of the current layer calculated based on the calculation result of the \((L+1)\) layer is as follows:

\[
g^L = (g^L_x, g^L_y).
\]

(1)

The residual light flux to be calculated in the actual \(L\) layer is as follows:

\[
d^L = (d^L_x, d^L_y).
\]

(2)

Redefine the error function as follows:

\[
\epsilon = \sum (I(x, y) - I(x + g^L_x + d^L_x, y + g^L_y + d^L_y)).
\]

(3)

The calculation method of the next layer \(g\) after obtaining \(g\) and \(d\) in the iterative process is as follows:

\[
g^{-1} = 2(g^L + d^L).
\]

(4)

The initial value of the iteration process is set to:

\[
g^{Lm} = [0 \ 0]^T.
\]

(5)

Before calculating the residual optical flow, first, use the \(g^L\) calculated in the previous layer to simplify the problem of two images \(A\) and \(B\):

\[
A(x, y) = I^A(x, y),
\]

(6)

\[
B(x, y) = I^B(x + g^L + d^L, y + g^L + d^L).
\]

(7)

The problem becomes to minimize the error as follows:

\[
\epsilon(V) = \epsilon(v_x, v_y) = \sum_{p=x-w_x}^{p=x+w_x} \sum_{y=y-w_y}^{y=y+w_y} (A(x, y) - B(x + v_x, y + v_y))^2.
\]

(8)

The objective function is optimized to solve for the motion vector \(V\) of feature point, and the minimum value should be at the stationary point, so should derive this function:

\[
\frac{\partial \epsilon(V)}{\partial v} = \begin{bmatrix} 0 & 0 \end{bmatrix}.
\]

(9)

And:

\[
\frac{\partial \epsilon(V)}{\partial (x, y)} = -2 \sum_{p=x-w_x}^{p=x+w_x} \sum_{y=y-w_y}^{y=y+w_y} \begin{bmatrix} \frac{\partial B}{\partial x} & \frac{\partial B}{\partial y} \end{bmatrix}.
\]

(10)
Expanding on B Taylor at point $P(x, y)$ as follows:

$$\frac{\partial (\Psi)}{\partial \Psi} \approx -2 \sum_{x, y} p_{x, y} \sum_{x, y} \left( A(x, y) - B(x, y) - \frac{\partial B}{\partial x} \frac{\partial B}{\partial y} \right) \Psi$$

Obtain:

$$\frac{1}{2} \left[ \frac{\partial (\Psi)}{\partial \Psi} \right]^T \approx G\Psi - B. \quad (11)$$

The stagnation point is 0, and then:

$$\Psi_{opt} = G^{-1}B. \quad (13)$$

Then the residual light flux is solved. Then the residual light flux is solved. ViBe+ is an image processing algorithm improved by Van et al. based on ViBe [21], which can build a background model by collecting background samples. The characteristics of ViBe+ algorithm are as follows: a sample set is stored for all pixels, and the sample value stored in the sample set is the pixel value of the last point and the pixel value of its neighboring points. Compare the new pixel value of each subsequent frame of the sample historical value in the sample set to determine whether it belongs to the background point. The assumption is that the background is stationary or an object that moves very slowly. Foreground is an object that is moving relative to the background object. Therefore, the background extraction algorithm can also be regarded as a classification problem, traversing pixels to determine whether a pixel belongs to a foreground background point. In the ViBe+ model, the background model is a sample set of 20 points stored per pixel. For a new frame of the image, when a certain pixel of the frame is relatively close to the sampling value in the sample set of the pixel, it can be judged as a background point.

Then differentiating $\nu(x, y)$ from all sample values in $M(x, y)$, and the number of all the differences that are in the range $\pm R$ is $N_b$. If $N_b$ is greater than a given threshold value min, it means that the current pixel value is similar to multiple values in the point’s history sample, then the point $(x, y)$ is considered to be a background point.

$$\nu(x, y): \text{pixel value at the (x, y) point.}$$

$$M(x, y) = \{v_1(x, y), v_2(x, y), \ldots, v_N(x, y)\}: \text{the background sample set at the pixel (x, y) (sample set size } N).$$

$R$: the upper and lower range of values.

During the construction of the 3D point cloud map, it is necessary to use the LK optical flow method and ViBe+ algorithm to dynamically detect continuous frame information in real-time, and the real-time update algorithm is shown in Algorithm 1.

A deep learning model will be introduced to complete the semantic segmentation when building the semantic map. So, in addition to utilizing the aforementioned dynamic detection methods, this paper makes full use of deep learning methods that will further eliminate potentially dynamic information while building the semantic map, such as common lifeforms. Figure 3 clearly expresses the changes in the feature points before and after the use of this dynamic elimination algorithm. Experiments show that the dynamic elimination algorithm in this paper can eliminate redundant noise feature points.

### 3. Object Detection and Instance Segmentation Algorithm Based on YOLACT

The task of object detection is to find all the objects of interest in the image and determines their geometric characteristics, which is the main task of robot vision. At present, the mainstream convolutional neural network Mask RCNN [22] and Mask Scoring RCNN [23], these two instance segmentation algorithms have higher precision, but lower recognition speed, while YOLO and Deeplab series of YOLO v4 [24], Deeplab v3+ [25] are faster in recognition speed, but are weaker than convolutional neural network series in segmentation precision. SOLO v2 [26] of the SOLO instance segmentation series, strikes a good balance between recognition precision and speed, but does not achieve real-time results under the requirement of high-precision segmentation. The experimental data show that the YOLACT
algorithm has the characteristics of fast recognition speed, high quality of segmentation mask, universal module, and strong universality. Therefore, this paper is based on the real-time instance segmentation YOLACT model design network. The frame diagram of the YOLACT model is shown in Figure 4:

YOLACT adds a mask branch of the existing One-stage detector to achieve the purpose of instance segmentation, but this process does not introduce a feature location step. YOLACT accomplishes this task by adding two parallel branches: the first uses the Fully Convolutional Network (FCN) to generate a series of prototype masks independent of a single instance, the second adds an extra head to the detection branch to predict the mask coefficients that can be used to encode the prototype masks with instance-specific weights. The final prediction is obtained in the Non-Maximum Suppression (NMS) step by linearly combining the output results from the two branches. Since the goal of the partitioning task is to obtain the mask, which is characterized by the existence of spatial connections, YOLACT uses the above organization. Most One-stage detectors predict box parameters and categories through the Fully Connected (FC) layer. Two-stage preserves spatial information on feature positioning steps such as Regions of Interest (ROI) align while outputting the mask using the Conv layer, but these operations must wait for Region Proposal Network (RPN) to complete, which greatly affects the segmentation efficiency. In YOLACT, the FC layer is responsible for predicting semantic tags, and the Conv layer is responsible for predicting the prototype mask and mask coefficients. The two branches are parallel, and finally assembled by matrix multiplication. So that, not only the spatial correlation is preserved, but also the One-stage model structure is maintained and the splitting speed is extremely fast.

**Algorithm 1: Dynamic information detection.**

```plaintext
Input: Get video streaming from the camera \( X_{N} \);
Output: Dynamic zone template Mask, Processed video stream \( Y \);
(1) Use the Lucas-Kanade method to detect dynamic areas;
(2) if \( Fun_{LK}(X_{\text{new}}) == \text{true} \) then
(3) (1) Intersect the same area as the previous frame to get \( X_{\text{now}} \);
(4) (2) Do ViBe+ dynamic detection of the current frame being intercepted with the previous frame
\( \text{Mask}, Y = Fun_{\text{ViBe}}(X_{\text{new}}, X_{\text{last}}) \);
(5) else
(6) Mask = None, \( Y = X_{\text{now}} \);
(7) end if
```

Figure 3: The performance of this paper in removing noise feature points in the TUM Dataset, (a) Raw image data. (b) Dynamic feature points are not culled. (c) Dynamic feature points are culled.

4. Construction of 3D Semantic Map

4.1. Formatting of Mathematical Components. This paper divides the process of VSLAM construction of semantic maps into three parts: data acquisition, data processing, and data storage. The acquisition of experimental data in this paper mainly relies on Kinect DK camera of Microsoft Corporation and on the official Dataset of TUM. In the second part and the third part of the paper, the data
processing work after obtaining the original data is introduced, the problem of interference caused by moving objects in the SLAM process is solved, and the search for a semantic segmentation algorithm that is robust in the real-time VSLAM process. The next section focuses on the construction of point cloud maps using MAPLAB [29]. The MAPLAB framework is mainly composed of two parts:

1. Online front-end ROVIOLI realizes Visual Inertial Odometry (VIO) and positioning. Receive image and inertial sensor data as input, output global pose estimation, and build Visual-Inertial (VI) map.
2. Offline MAPLAB console, allowing users to perform various algorithm processing on the map in offline batch processing, which can be used as a research and test platform for new algorithms.

MAPLAB mainly includes five commonly used functions: mapping and localization, multisession mapping, map maintenance, largescale mapping, and dense reconstruction. Compared with ORB-SLAM2 algorithm, MAPLAB has done a lot of work in the global pose estimation without drift, which enhances the precision of robot manipulation or navigation. MAPLAB also provides tools for dense reconstruction, either by optimizing feature points in a sparse map or by obtaining denser point clouds directly from RGB-D camera.

**4.2. Construction of the Semantic Database.** After acquiring data from RGB-D camera, first use the dynamic elimination algorithm in this paper to eliminate dynamic noise, then use YOLACT for 2D object detection, then use the projection method to extend 2D semantic information to the 3D data, and last construct a 3D semantics database. In the construction of the semantic database, the YOLACT model is mainly used to classify 3D objects observed in each frame and add it to the database. When the same kind of object is detected again in the next frame, the algorithm will determine whether it is an object that has already been recorded. If it has been recorded, it will optimize the object based on the new and old semantics and spatial pose information. If it has not been recorded, record the object directly in the database. The flow chart of the semantic object database construction is shown in Figure 5.

![Figure 4: Framework diagram of YOLACT [19].](image)

**Table 1: Comparison of speed and precision between YOLACT and other algorithms.**

| Method   | Backbone | FPS | Time  | AP   | AP50 | AP75 | APS  | APM  | APL  |
|----------|----------|-----|-------|------|------|------|------|------|------|
| PA-net [27] | R-50-FPN | 4.7 | 212.8 | 36.6 | 58.0 | 39.3 | 16.3 | 38.1 | 53.1 |
| FCIS [28]   | R-101-FPN | 6.6 | 151.5 | 29.5 | 51.5 | 30.2 | 8.0  | 31.0 | 49.7 |
| Mask R-CNN [21] | R-101-FPN | 8.6 | 116.3 | 35.7 | 58.0 | 37.8 | 15.5 | 38.1 | 52.4 |
| MS R-CNN [22] | R-101-FPN | 8.6 | 116.3 | 38.3 | 58.8 | 41.5 | 17.8 | 40.4 | 54.4 |
| YOLACT     | R-101-FPN | 38.6 | 24.6  | 28.5 | 46.9 | 29.9 | 9.5  | 30.1 | 45.5 |

4.3. Construction of Semantic Map. It is difficult to construct a semantic map with less noise information and strong usability in a complex dynamic environment. The above section proposes a dynamic elimination optimization algorithm, a semantic segmentation algorithm, and a semantic database construction algorithm based on the YOLACT deep learning model, optical flow method, and ViBe+. In this section, related algorithms will be combined to achieve a robust semantics Map construction algorithm.

In this paper, RGB-D cameras are used to obtain the coordinate information \((u, v)\) and the depth information \(d\). The 3D coordinates can be obtained by the following formula:

\[
[X, Y, Z]^T = \left[ \frac{(u - c_x)Z}{f_x}, \frac{(u - c_x)Z}{f_y}, \frac{d}{s} \right]^T.
\] (14)

Among them, \(f_x\) and \(f_y\) are the focal lengths of the camera on the \(x\) and \(y\) axes, respectively, and \(c_x\) and \(c_y\) are the translations between the pixel coordinates and the
imaging plane, which can be obtained through the camera’s internal parameter matrix, s is the zoom factor of the depth map. All the data for building the semantic map are thus obtained. The process of constructing the semantic map is as follows: Use this algorithm combining the optical flow method and ViBe+ to process the acquired RGB-D data stream. Send processed 2D image data to the YOLACT segmentation model to further complete the dynamic noise removal, obtain 2D semantic information, and then combine the 2D semantic information with the corresponding depth data. Update the semantic database and integrate the point cloud data into a complete semantic map using MAPLAB algorithm. The above algorithm can be seen in Algorithm 2.

5. Experimental Verification and Analysis

5.1. Introduction to the Experimental Platform. This experiment uses a self-built experimental platform, equipped with the latest Kinect DK depth camera from Microsoft, and uses TUM Datasets for training and testing. The deep learning training and the construction of semantic map are all carried out under Ubuntu 18.04 system environment. The processor model is Intel i9-9900k, with 64 GB memory. To get higher training and testing speed, this paper uses RTX2080Ti graphics card to accelerate training.

5.2. Training of YOLACT Model. To train YOLACT model, this paper collects Lab-Data and expands the data by randomly stretching, rotating, symmetrical, and increasing noise. 15 common objects are selected for labeling as shown in Table 2. Finally get 8000 picture data. Randomly select 85% of the Lab-Dataset for training and 15% for verification. The trained model is applied to real-time semantic segmentation, and the segmentation effect is shown in Figure 6.

The precision of the algorithm mentioned in this article is now theoretically analyzed. Assuming that when N objects of multiple categories are detected, the case of judging the correct object category is called a positive example, and the case of judging the wrong object category is a negative example, then the classification of an object can have the following four situations. It is actually a positive example and is judged to be a positive example, denoted as TP (true positive). It is actually a positive case but is judged as a negative case, which is recorded as FN (false negative). It is actually a negative case but is judged to be a positive case, which is recorded as FP (false positive). It is actually a negative case and is judged as a negative case, which is recorded as TN (true negative). Define Precision = TP/(TP + FP), Recall = TP/(TP + FN), use the Lab-Dataset to obtain several sets of Precision and Recall values by selecting different confidence thresholds, use precision as the ordinate and recall as the abscissa can get the $P-R$ curve. Figure 7 is obtained by comparing the $P-R$ curve obtained by the YOLACT algorithm selected in this paper with other algorithms. From the experimental curves, it can be seen that the YOLACT model trained in this paper has a significant advantage over the segmentation precision, which meets the needs of semantic map construction.

5.3. Construction of a 3D Map. Figure 8 compares the algorithm in this paper with the ORB-SLAM2 trajectory prediction curve. The solid line in the figure shows the position of the trajectory in the 3D space. The ground truth
**Input**: RGB-D data stream from the camera.  
**Output**: Semantic map.  
1. Pre-selection KeyFrame \((x, y)\).  
2. Use Algorithm 1 for Keyframe Processing  
   
   \[ Y = \text{Algorithm1}(\text{KeyFrame}(x, y)) \]  
3. Use Mask optimizing DepthImg to obtain DepthImg\text{new}  
   
   \[ \text{DepthImg}_{\text{new}} = \text{Fun}_{\text{mask}}(\text{pMask}, \text{DepthImg}) \]  
4. Use YOLACT to process \( Y \) to obtain 2D semantic information  
   
   \[ \text{Semantic}_{2D} = \text{YOLACT}(Y) \]  
5. Mapping 2D semantic information to 3D data  
   
   \[ \text{Semantic}_{3D} = \text{Fun}_{\text{2D}\rightarrow\text{3D}}(\text{Semantic}_{2D}, \text{DepthImg}_{\text{new}}) \]  
6. if \( \text{Fun}_{\text{find}}(\text{Database}, \text{Semantic}_{3D}) \neq \text{false} \) then:  
7. (1) Update database \( \text{Fun}_{\text{add}}(\text{Database}, \text{Semantic}_{3D}) \).  
8. (2) Building a semantic map \( \text{Map}_{\text{add}}(\text{Semantic}_{3D}). \)  
9. else:  
10. (1) Calculate the shortest distance between the nearest object  
    
    \[ \text{Distance}_{\text{short}} = \text{Fun}_{\text{distance}}(\text{Database}, \text{Semantic}_{3D}) \]  
11. (2) if \( \text{Distance}_{\text{short}} > \text{Distance}_{\text{max}} \) then:  
12. (3) Add the object to the semantic map \( \text{Map}_{\text{add}}(\text{Semantic}_{3D}) \).  
13. (3) else:  
14. (4) Optimize the semantic map \( \text{Map}_{\text{update}}(\text{Semantic}_{3D}) \).  

**Algorithm 2**: Update Algorithm for the construction of semantic maps.

| Chair | Backpack | Screen | Laptop | Person |  
|-------|----------|--------|--------|--------|  
| Book  | Drone    | Bottle | Robot  | Mouse  |  
| Door  | Switchbox| Tv     | Keyboard| Desk   |  

**Table 2**: 15 categories of objects.

![Figure 6: YOLACT’s inroom semantic recognition performance.](image)

illustrated by the green solid line represents the real trajectory recorded by the motion capture device, and ORB-SLAM2 illustrated by the solid blue line represents the trajectory obtained by the ORB-SLAM2 algorithm of the original data, and the solid red line represents the trajectory obtained by the algorithm proposed in this paper. In order to facilitate the comprehensive comparison of the positions between the trajectories, the 3D trajectory map is projected, and projected the 3D trajectory map is represented by a dotted line, and the color of the dotted line corresponds to the solid line.
Figure 7: PR curve based on Lab-Dataset.

Figure 8: 3D trajectory curve.

Figure 9: Continued.
Figure 9 shows the comparison data of this algorithm with orb-slam2 and ds-slam experiments on two sets of tum data sets. The absolute pose error (APE) of the camera trajectory is drawn and compared with the relative pose error (RPE). Among them, orb-slam2 is the data that have not been dynamically eliminated. It is obvious that there are large errors in both ape and RPE. Compared with ds-slam with a dynamic culling algorithm, this algorithm also has the
advantage of less error. The comparison of three SLAM algorithms shows that the motion cancellation algorithm proposed in this paper can effectively eliminate the dynamic information in the high motion experimental environment and improve the accuracy of VSLAM trajectory estimation.

Use the trained YOLACT model for semantic segmentation. Then the keyframe pictures are saved, the map points, semantic information, and the position information of the object in the keyframe corresponding to all the keyframe pictures are stored and the map is constructed by fusing these data. The 3D point cloud map construction effect and the semantic map construction effect before and after dynamic information rejection are shown in Figure 10, and Figure 10(a) shows the performance of the 3D point cloud map (Figure 10(a_1)), the 3D semantic map (Figure 10(a_2)) without the dynamic rejection algorithm, Figure 10(b) shows the performance of the 3D point cloud map (Figure 10(b_1)) and the 3D semantic map (Figure 10(b_2)) processed by the dynamic rejection algorithm.

Through experiments, it can be seen that the processed data have outstanding performance in both the construction of 3D maps and the construction of semantic maps. Compared with unprocessed data, the semantic map constructed has the advantages of less noise information and precise semantic information. Figure 11 shows the performance of the algorithm in the indoor dynamic environment. The figure shows the unused items corresponding to different colors. In the experiment, a total of 10 objects were identified, which were labeled with ten colors.

6. Conclusion

This paper mainly uses ViBe+ and optical flow method combined with a deep learning model to perform dynamic detection of mobile cameras and remove dynamic information, combined with YOLACT, MAPLAB, 3D semantic library update algorithm, etc. to construct a 3D semantic map. The main contribution is to propose a dynamic detection algorithm combining ViBe+ and optical flow method and introduce a deep learning model, which has a greater improvement in precision than the traditional optical flow method. From the comparative experiments, it is shown that using the dynamic information elimination algorithm proposed in this paper can reduce the error of computer estimation of camera position. And it reduces the impact of dynamic information on VSLAM in indoor environment, reduces the error of constructing a geometric map, eliminates some dynamic noise, and improves the robustness of the system. In addition, the constructed semantic map has few noisy point clouds and high map precision.

In the future work, we will focus on expanding the application scenarios, from the semantic construction of indoor small-scale dynamic environment to the larger environment. When this work is expanded from indoor to outdoor, how to deal with the sudden change of image brightness caused by the change of light intensity will become a major challenge to the dynamic culling work and an important research direction.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant nos. 61801323 and 61972454), China Postdoctoral Science Foundation (Grant no. 2021M691848), and Science and Technology Projects Fund of Suzhou (Grant nos. SS2019029 and SYG202142).

References

[1] A. Singandhupe and H. La, "A review of slam techniques and security in autonomous driving," in Proceedings of the 2019 Third IEEE International Conference on Robotic Computing (IRC), pp. 602–607, IEEE, Naples, Italy, February 2019.
[2] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, “Simultaneous localization and mapping: a survey of current trends in autonomous driving,” IEEE Transactions on Intelligent Vehicles, vol. 2, no. 3, pp. 194–220, 2017.
[3] R. Mur-Artal and J. D. Tardos, “Orb-slam2: an open-source slam system for monocular, stereo, and rgb-d cameras,” IEEE Transactions on Robotics, vol. 33, no. 5, pp. 1255–1262, 2017.

[4] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison, “Codeslam learning a compact, optimisable representation for dense visual slam,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2560–2568, 2018.

[5] C. Kerl, J. Sturm, and D. Cremers, “Dense visual SLAM for RGB-D cameras,” in Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2100–2106, IEEE, Tokyo, Japan, November 2013.

[6] H. Li, J. Yao, J. C. Bazin, X. Lu, Y. Xing, and K. Liu, “A monocular slam system leveraging structural regularity in manhattan world,” in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 2518–2525, IEEE, Brisbane, QLD, Australia, May 2018.

[7] J. Mccormac, A. Handa, and A. J. Davison, “SemanticFusion: dense 3D semantic mapping with convolutional neural networks,” International Conference on Robotics and Automation, pp. 4628–4635, 2017.

[8] F. Yu, J. Shang, Y. Hu, and M. Milford, “NeuroSLAM: a brain-inspired slam system for 3D environments,” Biological Cybernetics, vol. 113, no. 5-6, pp. 515–545, 2019.

[9] S. Li and D. Lee, “RGB-D SLAM in dynamic environments using static point weighting,” IEEE Robotics and Automation Letters, vol. 2, no. 4, pp. 2263–2270, 2017.

[10] N. Brasch, A. Boric, and J. Lallemant, “Semantic monocular SLAM for highly dynamic environments,” Intelligent Robots and Systems, pp. 393–400, 2018.

[11] G. Liu, W. Zeng, B. Feng, and F. Xu, “DMS-SLAM: a general visual SLAM system for dynamic scenes with multiple sensors,” Sensors, vol. 19, no. 17, p. 3714, 2019.

[12] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, and A. Davison, “MID-Fusion: octree-based object-level multi-instance dynamic slam,” in Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), pp. 5231–5237, IEEE, Montreal, QC, Canada, May 2019.

[13] H. Rebecq, T. Horstschaefer, G. Gallego, and D. Scaramuzza, “EVO: a geometric approach to event-based 6-DOF parallel tracking and mapping in real time,” IEEE Robotics and Automation Letters, vol. 2, no. 4, pp. 2263–2270, 2017.

[14] C. Yu, Z. Liu, X. J. Liu, F. Xie, Y. Yang, and Q. Wei, “DS-SLAM: a semantic visual SLAM towards dynamic environments,” in Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1168–1174, IEEE, Madrid, Spain, October 2018.

[15] N. D. Reddy, P. Singhal, V. Chari, and K. M. Krishna, “Dynamic body slam with semantic constraints,” in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1897–1904, IEEE, Hamburg, Germany, October 2015.

[16] D. Flores and G. Flores, “The STDyn-SLAM: a stereo vision and semantic segmentation approach for VSLAM in dynamic outdoor environments,” IEEE Access, vol. 10, pp. 18201–18209, 2022.

[17] M. R. U. Saputra, A. Markham, and N. Trigoni, “Visual SLAM and structure from motion in dynamic environments,” ACM Computing Surveys, vol. 51, no. 2, pp. 1–36, 2019.

[18] B. Bescos, J. M. Facil, J. Civera, and J. Neira, “DynaSLAM: tracking, mapping, and inpainting in dynamic scenes,” IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 4076–4083, 2018.

[19] D. Bolya, C. Zhou, and F. Xiao, “Yolact: real-time instance segmentation,” in Proceedings of the IEEE international conference on computer vision, pp. 9157–9166, 2019.

[20] V. M. Droogenbroeck and O. Paquot, “Background subtraction: experiments and improvements for vib,” in Proceedings of the 2012 IEEE computer society conference on computer vision and pattern recognition workshops, pp. 32–37, IEEE, Providence, RI, USA, June 2012.

[21] O. Barnich and M. Van Droogenbroeck, “ViBe: a universal background subtraction algorithm for video sequences,” IEEE Transactions on Image Processing, vol. 20, no. 6, pp. 1709–1724, 2011.

[22] H. Kaiming, G. Georgia, D. Piotr, and G. Ross, “Mask R-CNN: IEEE transactions on pattern analysis and machine intelligence,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 2, pp. 386–397, 2018.

[23] Z. Huang, L. Huang, and Y. Gong, “Mask scoring r-cnn,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 6409–6418, 2019.

[24] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection, http://arXiv.org/abs/2004.10934, 2020.

[25] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in Proceedings of the European conference on computer vision (ECCV), pp. 801–818, 2018.

[26] X. Wang, R. Zhng, T. Kong, L. Li, and C. Shen, “Solv2: dynamic and fast instance segmentation,” Advances in Neural Information Processing Systems, 2020.

[27] S. Liu, L. Qi, and H. Qin, “Path aggregation network for instance segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8759–8768, 2018.

[28] Y. Li, H. Qi, and J. Dai, “Fully convolutional instance-aware semantic segmentation,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2359–2367, 2017.

[29] T. Schneider, M. Dymczyk, M. Fehr et al., “maplab: an open framework for research in visual-inertial mapping and localization,” IEEE Robotics and Automation Letters, vol. 3, no. 3, pp. 1418–1425, 2018.