DYP-SLAM: A Real-time Visual SLAM Based on YOLO and Probability in Dynamic Environments

Xinggang Hu\textsuperscript{1}, Yunzhou Zhang\textsuperscript{1*}, Zhenzhong Cao\textsuperscript{1}, Yanmin Wu\textsuperscript{2}, Zhiqiang Deng\textsuperscript{1}, Wenkai Sun\textsuperscript{1}, Sonya Coleman\textsuperscript{2}, Dermot Kerr\textsuperscript{3}

Abstract—SLAM algorithm is based on the static assumption of environment. Therefore, the dynamic factors in the environment will have a great impact on the matching points due to violating this assumption, and then directly affect the accuracy of subsequent camera pose estimation. Recently, some related works generally use the combination of semantic constraints and geometric constraints to deal with dynamic objects, but there are some problems, such as poor real-time performance, easy to treat people as rigid bodies, and poor performance in low dynamic scenes. In this paper, a dynamic scene oriented visual SLAM algorithm based on target detection and static probability named DYP-SLAM is proposed. The algorithm combines semantic constraints and geometric constraints to calculate the static probability of objects, keypoints and map points, and takes them as weights to participate in camera pose estimation. The proposed algorithm is tested on the public dataset and compared with a variety of advanced algorithms. It has achieved the best results in almost all low dynamics and high dynamic scenarios, and showing quite high real-time.

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) is the key technology for autonomous navigation of mobile robots, and it is widely applied in the fields of autopilot, UAV and augmented reality (AR). SLAM system is based on environmental static assumption \cite{1}, and dynamic factors will bring wrong observation data to the system, making it difficult to establish various geometric constraints on which SLAM system works, and reducing the accuracy and robustness of SLAM system. The abnormal point processing mechanism of RANSAC (Random Sample Consensus) algorithm can solve the influence of some dynamic points in static or low dynamic environment. However, when dynamic objects occupy most of the environment, RANSAC algorithm has little effect.

With the development of deep learning technology, some advanced researchers in recent years have used semantic constraints to solve the visual SLAM problem in dynamic environment. The general approach is to take the semantic information obtained from semantic segmentation as a priori and eliminate the dynamic objects in the environment combined with geometric constraints. However, there are still some problems in these methods. First, the calculation cost of semantic segmentation network represented by mask R-CNN \cite{2} is very high \cite{3}, while the segmentation accuracy of lightweight network represented by SegNet \cite{4} is difficult to guarantee. Second, the segmentation boundary of the current semantic segmentation network is not accurate. Third, as non-rigid bodies, people often have local motion. Directly eliminating them as a whole will reduce the constraints of feature points and affect the positioning accuracy. Fourth, the current scheme treats the dynamic environment as a high dynamic attribute, resulting in poor performance in low dynamic scenes.

For the above problems, we proposed DYP-SLAM, which is a high-performance high-efficiency visual SLAM system based on target detection and static probability in indoor dynamic environments. On the basis of ORB-SLAM2 \cite{5}, DYP-SLAM uses YOLOv5 to obtain semantic information, uses extended Kalman filter (EKF) and Hungarian algorithm to compensate missed detection, distinguishes foreground points and background points of target detection results based on DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm, and calculates the static probability of objects to distinguish high dynamic objects from low dynamic objects. Then, the static probability of keypoints is calculated and updated based on the projection constraints and the epipolar constraints, which is used as the weight to participate in the camera pose optimization. In addition, without calculating the static probability of objects, we provide a lower performance version to improve the real-time performance to meet the needs of different scenarios.

The main contributions of this paper are as follows:

- Compensate for missed detection based on EKF and Hungarian algorithm, improve the recall rate, and then improve the stability of visual SLAM in dynamic scene.
- Based on the YOLOv5 target detection results and geometric constraints, the static probability of potential moving objects is calculated, and the object motion attributes are divided into high dynamics and low dynamics, which are provided to the subsequent methods as a priori information for processing with different strategies, so as to improve the robustness and adaptability of SLAM system.
- A two-stage static probability of keypoints calculation method based on the DBSCAN clustering algorithm, the epipolar constraints and the projection constraints

\textsuperscript{*}The corresponding author of this paper.

\textsuperscript{1}Xinggang Hu, Yunzhou Zhang, Zhenzhong Cao, Zhiqiang Deng and Wenkai Sun are with College of Information Science and Engineering, Northeastern University, Shenyang 110819, China (Email: zhangyunzhou@mail.neu.edu.cn).

\textsuperscript{2}Yanmin Wu at Peking University, Shenzhen, China.

\textsuperscript{3}Sonya Coleman and Dermot Kerr are with School of Computing, Engineering and Intelligent Systems, Ulster University, N. Ireland, UK.

This work was supported by National Natural Science Foundation of China (No. 61973066, 61471110) and the Distinguished Creative Talent Program of Liaoning Colleges and Universities (LR2019027).
is proposed to solve the problem of false deletion of static keypoints caused by non-rigid body only local motion.

- We have compared with a variety of most advanced methods in multiple datasets. Experiments show that our algorithm can achieve almost the best results in high dynamic and low dynamic scenarios.

II. RELATED WORK

A. Dynamic SLAM without a Priori Semantic Information

When there is no semantic information as a priori, using reliable constraints to find the correct feature matching relationship is the basic method to deal with dynamic SLAM problem. [6] proposes a static weighting method of keyframe edge points, and integrated into the IAIACP method to reduce tracking error. [7] roughly detects the motion of moving objects based on self motion compensation image difference, and enhances the motion detection by tracking the motion using particle filter. [8] distinguishes dynamic and static map points based on feature correlation. DMS-SLAM [9] uses GMS [10] to eliminate mismatched points. [11] proposes a dense visual mileage calculation method based on background model to estimate the nonparametric background model from depth scene. Flowfusion [12] uses optical flow residuals to highlight dynamic regions in rgbd point clouds. Because there is no need for deep learning networks to provide semantic priors, the above methods are usually fast in dealing with dynamic factors in the environment, but lack of accuracy.

B. Dynamic SLAM Based on Semantic Constraints

Semantic segmentation or target detection can provide a steady and reliable priority constraint for dynamic SLAM. Detect-SLAM [13] detects targets in keyframes and propagates the motion probability of keypoints in real time to eliminate the influence of dynamic targets in SLAM. DS-SLAM [14] judges people’s dynamic and static attributes based on semantic information provided by SegNet, combined with sparse optical flow and motion consistency detection. Dyna- SLAM [15] combines mask R-CNN [2] and multi view geometry to process moving objects. Dynamic-SLAM [16] compensates SSD for missed detection based on the speed invariance of adjacent frames, and eliminates dynamic objects combined with selective tracking algorithm. SaD-SLAM [17] extracts static feature points from objects judged as dynamic by mask R-CNN by verifying whether the inter frame feature points meet the epipolar constraints. DP-SLAM [18] updates the priori motion probability of semantic information and the posteriori motion probability obtained by epipolar geometric constraints by Bayes to obtain the final motion probability of feature points. Blitz-SLAM [19] repairs the mask of BlitzNet based on depth information, and classifies static and dynamic matching points in potential dynamic areas using epipolar constraints. [20] only performs semantic segmentation on keyframes, clusters the depth map and identifies moving objects combined with re-projection error to remove known and unknown dynamic objects. Generally, the above methods can accurately eliminate dynamic objects in the environment, but it is difficult to give consideration to both positioning accuracy and real-time, and the performance is generally poor in low dynamic scenes.

III. SYSTEM OVERVIEW

A. Definition of Variables

In this paper, common variables are defined as follows:

- $F_k$ - Frame K
- $(\cdot)_w$ is the world coordinate, $(\cdot)_k$ is the camera coordinate of $F_k$, and $(\cdot)_w$ is the pixel coordinate.
- $K$ - The intrinsic matrix of a pinhole camera model.
- $T_{k,w}$ - The transformation from world frame to camera frame K, which is composed of a rotation $R_{k,w} \in R^{3 \times 3}$ and a translation $t_{k,w} \in R^{3 \times 1}$.
- $T_{k,1}$ - The transformation from camera frame K-1 to camera frame K.
- $P_i^k$ - The keypoint with ID $i$ in $F_k$. Its coordinate in the pixel coordinate is $P_{i,x}^k = [u_i, v_i]^T$, in the camera coordinate is $P_{i,c}^k = [X_i^k, Y_i^k, Z_i^k]^T$, in the world coordinate is $P_{i,w}^k = [X_i^k, Y_i^k, Z_i^k]^T$. $(\bullet)$ is the form of homogeneous coordinates in each coordinate system.
- $T_{i,x}^{k-1}$ - The keypoint with ID $i^+$ in $F_{k-1}$ which forms a matching relationship with $P_i^k$.
- $LP_i^k, LP_i^{k-1}$ - The matching point pair obtained by the optical flow tracking, which have the same coordinate definition as $P_i^k$ in each coordinate system.
- $OSP_{i,+}$ - The static probability of potential moving object with ID $i^+$, containing the keypoint with ID $i$ in $F_k$.
- $OSP_{Th}$ - The threshold to distinguish whether the object motion attribute is high dynamic or low dynamic, which is set to 0.9 in this paper.
- $KSP_{i,k}^{P^k}$ - The static probability of $P_i^k$, which is in the update state and participates in camera pose optimization.
- $KSP_{i,Tr}^{D^k}, KSP_{i,T}^{Tk}, KSP_{i,T}^{Fk}$ - The static probability of $P_i^k$ obtained by the DBSCAN clustering algorithm, the projection constraints and the epipolar constraints respectively.
- $C_{Tk}, C_{Tk}$ - The confidence of projection constraints and epipolar constraints in $F_k$.
- $MSP_{i,-}$ - The static probability of the map point forming a matching relationship with $P_i^k$.

B. System Architecture

Based on ORB-SLAM2 [5], we design a complete static probability calculation and update framework of keypoints based on multiple constraints to deal with the influence of moving objects in dynamic environment. The system obtains semantic information based on YOLOv5, compensates for missed detection based on EKF and Hungarian algorithm, and then associates data between adjacent frames. In $F_k$, only calculate and update the static probability of the keypoints inside the potential moving object box. Firstly, the static
probability of potential moving object $OSP^k_{i+}$ is obtained by using optical flow and the epipolar constraints, and the object is divided into high dynamic object and low dynamic object. Initialize $KSP^k_i$ as the static probability of the object to which the keypoint belongs. Then, the $KSP^D_{i+k}$ is calculated by using the DBSCAN clustering results, and the $KSP^k_i$ is updated to estimate the camera pose in the first stage to obtain $T_{k,x}$. Next, $KSP^F_{i+k}$, $KSP^R_{i+k}$, $C^F_{i+k}$, $C^R_{i+k}$ are obtained by using the projection constraints and the epipolar constraints, $KSP^k_i$ and $MSP^k_i$ are updated to participate in camera pose optimization as weights to obtain a more accurate $T_{k,x}$.

IV. SPECIFIC IMPLEMENTATION

A. Missed Detection Compensation Algorithm

When processing dynamic objects, if the semantic information as a priori is suddenly missing in some frames, on the one hand, the subsequent methods based on semantic priors will not be able to process dynamic objects. On the other hand, the sudden emergence of dynamic objects in high dynamic scenes will lead to a sharp increase in the number of keypoints incorrectly matched between adjacent frames, which leads to the loss of tracking in SLAM system in high dynamic scenario. Therefore, stable and accurate semantic information is critical.

In order to solve the missed detection problem of YOLOv5, we use EKF and Hungarian algorithm to compensate the missed detection of potential moving objects. EKF is used to predict the boxes of potential moving objects in $F_{k}$, and the Hungarian algorithm is used to correlate the predicted boxes with the boxes detected by YOLOv5. If the predicted box does not find a matching detected box, it can be considered that $F_{k}$ has missed detection, and the prediction result of EKF is used to compensate the missed detection result. After missed detection compensation, EKF and Hungarian algorithm are used again for inter frame data association of boxes.

B. DBSCAN Density Clustering Algorithm

Compared with semantic segmentation methods, target detection technology has great advantages in real-time, but it cannot provide accurate object mask. In the indoor dynamic SLAM scene, this problem leads to numerous static backgrounds in the boxes classified as people, and the false deletion of static keypoints will reduce the constraints of camera pose optimization and reduce the accuracy of camera pose estimation. We note that the foreground person in the box is a non-rigid body, its depth has good continuity, and usually has a large fault with the background depth. To this end, we use the DBSCAN density clustering algorithm to distinguish between the foreground and background points of boxes classified as people. The number of keypoints to be clustered in a box with human category is $n$, and the sample set is the depth set $DE = \{d_1, d_2, \cdots, d_n\}$ of keypoints. Minkowski distance is usually used to measure the distance of m-dimensional samples $x_i, x_j$:

\[
\text{dis}(x_i, x_j) = \left( \sum_{k=1}^{m} \left| x_{i,k} - x_{j,k} \right|^p \right)^{\frac{1}{p}}
\]

Because clustering is based on depth here, $k = 1, p = 1$ in Eq[1]. We adaptively determine the neighborhood radius $\epsilon$ of DBSCAN density clustering algorithm and the threshold of the number of samples in the neighborhood $minPts$. The specific process of DBSCAN density clustering algorithm is shown in Alg[2]. After clustering, the one with the lowest average value of samples in cluster $C = \{C_1, C_2, \cdots, C_k\}$ is taken as the foreground points of box.

C. Calculation and Update of Static Probability

1) Object Static Probability: When calculating the static probability of each potential moving object, we use the idea of DS-SLAM [14] for reference to solve the fundamental matrix $LF_{k, k-1}$ and get the polar error $Ld_{i+}^{F, k, k-1}$. We use the epipolar constraints and chi-square distribution to test the epipolar error. Since $LP_{iuv}^k$ has $k = 2$ degrees of freedom, if $LP_{iuv}^k$ is assumed to follow the Gauss Distribution $N(0, 1)$, then according to the chi-square distribution:

\[
\chi^2 q(x; k) = \begin{cases} 
\frac{2^{k/2} \Gamma(\frac{k}{2})}{\Gamma(\frac{k}{2})}, & x > 0 \\
0, & x \leq 0
\end{cases}
\]

The definition of the function $\Gamma(v)$ is:
Algorithm 1: Foreground and Background Keypoints Distinction Based on DBSCAN

**Input:** \( DE = (d_1, d_2, \ldots, d_n) \), \( \text{eps}, \minPts \), \( \text{dis}(d_i, d_j) \).

**Output:** Cluster partition \( C = \{C_1, C_2, \ldots, C_k\} \).

1. **Initialize:** set of core sample \( \Psi = \emptyset \), number of clusters \( k = 0 \), set of unaccessed samples \( \Pi = DE \), cluster partition \( C = \emptyset \).
2. for \( d_i \in DE \) do
   3. find the eps-neighborhood sub sample set \( N_{\text{eps}}(d_i) \) of sample by \( \text{dis}(d_i, d_j) \);
   4. if \( |N_{\text{eps}}(d_i)| \geq \minPts \) then
      5. \( \Psi = \Psi \cup \{d_i\} \);
   end
7. end
8. while \( \Psi \neq \emptyset \) do
   9. random select \( \phi \in \Psi \);
10. **Initialize:** Current cluster core sample queue \( \Psi_c = \{\phi\} \), Current cluster sample collection \( C_k = \{\phi\} \);
11. \( k = k + 1 \);
12. \( \Pi = \Pi \setminus \{\phi\} \);
13. while \( \Psi_c \neq \emptyset \) do
   14. select \( \phi^* \in \Psi_c \), which is the first core sample in \( \Psi_c \);
   15. \( \delta = N_{\text{eps}}(\phi^*) \cap \Pi \);
   16. \( C_k = C_k \cup \delta \);
   17. \( \Pi = \Pi \setminus \delta \);
   18. \( \Psi_c = \Psi_c \cup (\delta \cap \Psi) - \phi^* \);
end
20. \( C = C \cup C_k \);
21. \( \Psi = \Psi \setminus C_k \);
22. end
23. return \( C = \{C_1, C_2, \ldots, C_k\} \);

\[
\Gamma(v) = \int_0^\infty e^{-t^v} dt, \text{Re} v > 0 \quad (3)
\]

The single estimation result of \( OSP_{i,v}^k \) can be obtained:

\[
(OSP_{i,v}^k)_m = \text{clip} \left( \left( L_d^{k,F_{i,k-1}} \right)^2 ; 2 \right) \quad (4)
\]

After all estimation results are obtained by using all optical flow point pairs belonging to the object, all estimation results are sorted from small to large. Let the number of all estimation results be \( M \), and take the average value of \( (OSP_{i,v}^k)_m \) at 0.1M, 0.2M, 0.3M position after ranking as the estimated value of object static probability \( OSP_{i,v}^k \):

\[
OSP_{i,v}^k = \frac{(OSP_{i,v}^k)_{0.1M} + (OSP_{i,v}^k)_{0.2M} + (OSP_{i,v}^k)_{0.3M}}{3} \quad (5)
\]

According to the set object static probability threshold \( OSP_{Th} \), the object motion attributes are divided into high dynamic and low dynamic, which are provided to the subsequent methods as a priori information for processing with different strategies. The static probability of all keypoints in the box of the potential moving object is initialized to \( OSP_{i,v}^k \), and the static probability of other keypoints is initialized to 1.0.

2) **DBSCAN Clustering Algorithm:** After getting the DBSCAN clustering results, we adopt a soft strategy to further estimate the static probability of background points in the box of a potential moving object. Obviously, the static probability of background points must be greater than that of the object, and it is positively correlated with the static probability of the object. Specifies that the static probability of background points derived from the DBSCAN cluster is:

\[
KSP_{i,v}^{Dk} = \begin{cases} 
1 - \frac{OSP_{Th}^k}{OSP_{Th}^i} & (KSP_{i,v}^k) + 1, OSP_{i,v}^k \leq OSP_{Th}^i \\
\frac{1}{KSP_{i,v}^{Dk}} & OSP_{i,v}^k > OSP_{Th}^i
\end{cases} \quad (6)
\]

Considering that the static probability estimation of keypoints has not been strictly calculated at each point, and the camera pose estimation is vulnerable to dynamic points, we set the static probability of all foreground points in the box of high dynamic objects to 0.

3) **First Stage Camera Pose Optimization:** Update the static probability of keypoints:

\[
KSP_{i,v}^{Dk} = KSP_{i,v}^{Dk} \times KSP_{i,v}^{Dk} \quad (7)
\]

When initializing the SLAM system, map points will be created. At this time, the static probability of map point \( MSP_{i,v}^k \) will be initialized to the static probability of corresponding keypoint \( KSP_{i,v}^k \). In the frame after initialization, \( KSP_{i,v}^k \) and \( MSP_{i,v}^k \) are used as weights to optimize the camera pose, and the camera pose estimation value \( T_{k,w} \) in the first stage is obtained. Then, the static probability of \( P_{i,v}^k \), which has a matching relation with the keypoints in \( F_{k-1} \), is calculated based on the projection constraints and the epipolar constraints.

4) **Static Probability Based on the Projection Constraints:** Transform the \( P_{i,k-1} \) in \( F_{k-1} \) to \( \bar{c}_{k-1} \):

\[
P_{k-1}^{T_{i,v}} \cdot \frac{1}{K} \cdot \bar{P}_{i,v}^{k-1} = \frac{Z_{k-1}^{i,v}}{P_{i,v}^{k-1}} \quad (8)
\]

Transform and project \( P_{k-1}^{T_{i,v}} \) to \( F_k \), and the Euclidean distance between the projection point and \( P_{i,v}^k \) is:

\[
d_{i,v}^T = \left\| P_{i,v}^{k-1} - \frac{1}{T_{k-1}^{i,v}} \cdot \left| \begin{array}{c} 1 \\ T_{k-1}^{i,v} \end{array} \right| \right\|_{XYZ,w} \quad (9)
\]

Where function \( |P|_z \) represents the z-axis coordinate of point \( P \), and \( |P|_{XYZ,w} \) represents the non-homogeneous coordinate form of point \( P \). On the premise that the camera pose estimation \( T_{k,w} \) is relatively accurate, the greater \( d_{i,v}^T \), the greater the possibility that \( P_{i,v}^k \) and \( P_{i,v}^{k-1} \) are mismatched.
Based on this principle, we design a static probability model based on the projection constraints. After sorting the $d_i^T$ of all keypoints outside the box of the dynamic object in $F_k$ from small to large, take $d_i^T$ at the truncated position of 0.8 as the adaptive threshold $D_{Th}^F$ of the projection error, and obtain the minimum value $d_{min}$ of $d_i^T$. We use the Sigmoid function form to measure the static probability of keypoints of the matching relationship in the box:

$$KSP_{KSP}^{T} = \frac{1}{1 + e^{(d_i^T - D_{Th}^F) \times \frac{5}{D_{Th}^F - d_{min}^T}}}$$  \tag{10}

For a pair of matching points, the satisfaction of the projection constraints is not only related to whether the corresponding spatial points strictly meet the static environment assumption, but also directly related to the number of constraints when solving the pose matrix and whether the pose matrix itself is correctly solved. Therefore, the statistical confidence $C_i^{T}$ and calculation confidence $C_i^{C}$ of the pose matrix are introduced:

$$C_i^{T} = \frac{1}{1 + e^{-N_{BA} + 0.5ThBA}}$$  \tag{11}

$$C_i^{C} = 1 - \frac{\sum d_i^T}{NT \times D_{Th}^F}$$  \tag{12}

Where $N_{BA}$ is the number of interior points obtained by participating in the last camera pose solution, and $ThBA$ is the set minimum number of interior points required to participate in the camera pose solution, $NT$ and $\sum d_i^T$ respectively represent the number of all sample points and the sum of $d_i^T$ satisfying $d_i^T < D_{Th}^F$.

5) Static Probability Based on the Epipolar Constraints:

Based on the camera pose estimation $T_{k,w}$ in the first stage, a more accurate Fundamental matrix can be calculated:

$$F_{k,k-1} = K^{-T} (t_{k,k-1})^\wedge R_{k,k-1} K^{-1}$$  \tag{13}

The pole line $l_i = [A_i^k, B_i^k, C_i^k]^T$ corresponding to $P_i^k$ is:

$$l_i = F_{k,k-1} P_{k-1}^{k-1, u}$$  \tag{14}

Then the polar error $d_i^F$ is:

$$d_i^F = \sqrt{(A_i^k)^2 + (B_i^k)^2}$$  \tag{15}

Similar to the projection constraints, we design a method to calculate static probability and confidence based on the epipolar constraints:

$$KSP_i^{F} = \frac{1}{1 + e^{(d_i^F - D_{Th}^F) \times \frac{5}{D_{Th}^F - d_{min}^F}}}$$  \tag{16}

$$C_i^{F} = C_i^{T} = \frac{1}{1 + e^{-N_{BA} + 0.5ThBA}}$$  \tag{17}

$$C_i^{F} = 1 - \frac{\sum d_i^F}{NP \times D_{Th}^F}$$  \tag{18}

Where, $D_{Th}^F$ represents the adaptive threshold of epipolar error, $d_{min}^F$ represents the minimum value of $d_i^F$, $NP$ and $\sum d_i^F$ respectively represent the number of all sample points and the sum of $d_i^F$ satisfying $d_i^F < D_{Th}^F$.

It should be noted that, as can be seen from the Eq[13], the fundamental matrix cannot be obtained when the camera translation is not large enough. Therefore, when the camera translation is less than the set threshold $t_{Th}$, skip the calculation of static probability and confidence based on the epipolar constraints, that is:

$$KSP_i^{F} = 0, C_i^{F} = C_i^{T} = 0 \quad \text{Re} \parallel t_{k,k-1} \parallel_2 \leq t_{Th}$$  \tag{19}

6) Second Stage Camera Pose Optimization: After calculating the static probability of the keypoints based on the projection constraints and the epipolar constraints, we update the static probability of $P_i^k$ which matches the keypoints in $F_{k-1}$ for the second time. When the object is in high dynamics, the negative impact of dynamic points on camera pose estimation is generally greater than the positive impact of the increase in the number of static point constraints, which is just the opposite when the object is in low dynamics. So, when $OSP_{i^+} \leq OSP_{Th}$,

$$KSP_i^{k} = \left\{ \begin{array}{ll}
    KSP_i^{T} \times KSP_i^{F}, & \parallel t_{k,k-1} \parallel_2 > t_{Th} \\
    KSP_i^{T}, & \parallel t_{k,k-1} \parallel_2 \leq t_{Th}
\end{array} \right.$$  \tag{20}

when $OSP_{i^+} > OSP_{Th}$,

$$KSP_i^{k} = \frac{KSP_i^{T} \times C_i^{C}}{C_i^{C} + KSP_i^{F} \times C_i^{C}} + \frac{KSP_i^{F} \times C_i^{F}}{C_i^{F} + C_i^{F}}$$  \tag{21}

After missed detection compensation, we use EKF and Hungarian algorithm to correlate the boxes of potential moving objects between adjacent frames. It is easy to know that if the association result of a box in $F_k$ is not found in $F_{k-1}$, even if there is a matching relationship between the foreground points in the box, it is generally a false matching, so let $KSP_{i}^{k} = 0$ in this case. For $P_i^k$ that does not match the keypoints in $F_{k-1}$, according to the results of DBSCAN clustering, if $P_i^k$ belongs to the foreground points, let $KSP_{i}^{k} = 0$, else let $KSP_{i}^{k} = MSP_i$. After the second estimation result of $KSP_{i}^{k}$ is obtained, $MSP_i$ is updated. When $MSP_i < 0.3$, delete the map point. Then $KSP_i^{F}$ and $MSP_i$ are used as weights to participate in the second stage of camera pose optimization. When there is a big difference between $KSP_i^{F}$ and $MSP_i$, it can be considered that $KSP_i^{F}$ and $MSP_i$ are mismatched and do not participate in optimization.

V. EXPERIMENTS AND RESULTS

In this section, we test the performance of the proposed algorithm in 8 dynamic sequences of the TUM RGB-D
dataset [21], including 4 low dynamic sequences (fr3/s for short) and 4 high dynamic sequences (fr3/w for short), and the camera includes 4 kinds of motion: static, xyz, halfsphere and rpy. The indicators used to evaluate the accuracy are the Absolute Trajectory Error (ATE) and the Relative Pose Error (RPE). ATE represents the global consistency of trajectory. RPE includes translation drift and rotation drift. The Root-Mean-Square-Error (RMSE) and Standard Deviation (S.D.) of both are used to represent the robustness and stability of the system. Firstly, we compare with a variety of most advanced methods, then design ablation experiments to test the impact of each module, and finally carry out real-time analysis. All the experiments are performed on a computer with Intel i7 CPU, 3060 GPU, and 16GB memory.

A. Missed Detection Compensation and DBSCAN Clustering

In the dynamic SLAM scene, the motion of the object, the incomplete appearance of the object to be detected in the camera field of view, the blurred image and the singular angle of view caused by camera rotation all bring severe challenges to the target detection, very easy to cause miss detection, even will lead to continuous frame miss detection. Fig.2(a)-(d) and Fig.2(e) show the results of missed detection compensation in the above four cases and six consecutive frames, respectively. Fig.3 shows the DBSCAN clustering results after missed detection compensation. We selected two consecutive frames to show the clustering effect. The foreground points are marked with red and the background points are marked with green. The upper image group contains two people sitting on the chair and moving slowly respectively, and the people in the lower image group are in the fast walking state. The experimental results fully show the effectiveness and robustness of the missed detection compensation algorithm and clustering algorithm.

B. Comparison with State-of-the-arts

We contrast with ORB-SLAM2 [5] and forth most advanced dynamic SLAM methods, including DS-SLAM [14], Dyna-SLAM [15], Blitz-SLAM [19] and TRS [20]. Like our method, these algorithms are all improved based on ORB-SLAM2 [5]. Without calculating the static probability of the object, we provide a lower performance version of the algorithm in this paper to improve the real-time performance to meet the needs of different scenes, which is called DYP-SLAM−. The quantitative comparison results are shown in Tables I, II and III, in which the best results are highlighted.
TABLE II
RESULTS OF METRIC TRANSLATIONAL DRIFT (RPE)

| Sequences  | ORB-SLAM | Dyna-SLAM | DS-SLAM | Blitz-SLAM | TRS | DYP-SLAM−2 | DYP-SLAM−3 | DYP-SLAM−4 | DYP-SLAM−5 |
|------------|----------|-----------|---------|------------|-----|------------|------------|------------|------------|
| fr3/s/xyz  | 0.0117   | 0.0060    | 0.0142  | 0.0073     | /   | /          | /          | 0.0144     | 0.0071     |
| fr3/shalf  | 0.0231   | 0.0163    | 0.0239  | 0.0120     | /   | /          | /          | 0.0165     | 0.0073     |
| fr3/static | 0.0090   | 0.0043    | /       | /          | /   | /          | /          | /          | /          |
| fr3/rpy    | 0.0245   | 0.0144    | /       | /          | /   | /          | /          | /          | /          |
| fr3/w/xyz  | 0.3944   | 0.2964    | 0.0217  | 0.0119     | 0.0333 | 0.0229 | 0.0197 | 0.0096 | 0.0234 |
| fr3/w/half | 0.3480   | 0.2859    | 0.0284  | 0.0149     | 0.0297 | 0.0152 | 0.0253 | 0.0123 | 0.0423 |
| fr3/w/static| 0.2349   | 0.2151    | 0.0089  | 0.0044     | 0.0102 | 0.0048 | 0.0129 | 0.0069 | 0.0117 |
| fr3/w/rpy  | 0.4582   | 0.3447    | 0.0448  | 0.0262     | 0.1503 | 0.1168 | 0.0473 | 0.0283 | 0.0471 |

TABLE III
RESULTS OF METRIC ROTATIONAL DRIFT (RPE)

| Sequences  | ORB-SLAM | Dyna-SLAM | DS-SLAM | Blitz-SLAM | TRS | DYP-SLAM−2 | DYP-SLAM−3 | DYP-SLAM−4 | DYP-SLAM−5 |
|------------|----------|-----------|---------|------------|-----|------------|------------|------------|------------|
| fr3/s/xyz  | 0.4890   | 0.2713    | 0.5042  | 0.2651     | /   | /          | /          | 0.5024     | 0.2634     |
| fr3/shalf  | 0.6015   | 0.2924    | 0.7045  | 0.3488     | /   | /          | /          | 0.5981     | 0.2739     |
| fr3/static | 0.2850   | 0.1241    | /       | /          | /   | /          | /          | 0.2735     | 0.1215     |
| fr3/rpy    | 0.7722   | 0.3999    | /       | /          | /   | /          | /          | 0.8303     | 0.4653     |
| fr3/w/xyz  | 7.8466   | 5.8335    | 0.6284  | 0.3848     | 0.8266 | 0.5826 | 0.6132 | 0.3348 | 0.6368 |
| fr3/w/half | 7.2138   | 5.8299    | 0.7842  | 0.4012     | 0.8142 | 0.4101 | 0.7879 | 0.3751 | 0.9650 |
| fr3/w/static| 4.1856   | 3.8077    | 0.2612  | 0.1259     | 0.2690 | 0.1182 | 0.3058 | 0.1437 | 0.2872 |
| fr3/w/rpy  | 8.8923   | 6.6658    | 0.9894  | 0.5701     | 3.0042 | 2.3065 | 1.0841 | 0.6668 | 1.0587 |

Fig. 4. ATE and RPE from DYP-SLAM.

C. Ablation Experiment

In order to prove the function of each module of our algorithm, we designed an ablation experiment, and the experimental results are shown in Table IV. Among them, DYP-SLAM: The algorithm of this paper; DYP-SLAM−: Do not use object static probability; DYP-SLAM−2: No missed detection compensation; DYP-SLAM−3: DBSCAN clustering is not performed; DYP-SLAM−4: The static probability is not used, that is, all the foreground keypoints after missed detection compensation and DBSCAN clustering are directly eliminated; DYP-SLAM−5: Directly eliminate all keypoints in the box with human category. The experimental results show that DYP-SLAM− shows worse performance in low dynamic scenes because we cannot distinguish between high dynamic objects and low dynamic objects, so all objects are processed according to high dynamic; DYP-SLAM−2 is almost unaffected in low dynamic scenes, but the performance is very poor in w/rpy when the camera and object are moving violently; DYP-SLAM−3 and DYP-SLAM−4 show general performance in all sequences; DYP-SLAM−5 encounters difficulties in initialization due to insufficient features in almost all sequences, and tracking is lost in some sequences.

D. Real-time Analysis

Real-time performance is one of the important evaluation indexes of SLAM system. We tested the average running
time of each module, as shown in Table V. EKF represents the missed detection compensation and data association of boxes module, OSP represents the object static probability calculation module, and KSP represents the static probability calculation module of keypoints based on the epipolar constraint and the projection constraints. Semantic threads based on YOLOv5s run in parallel with ORB feature extraction. The results show that the average processing time per frame for the main threads of DYP-SLAM and DYP-SLAM− is 42.7 ms and 24.77 ms, that is, the running speed reaches 23fps/s and 40fps/s respectively. Compared with the dynamic SLAM algorithm based on semantic segmentation, it can better meet the real-time requirements.

TABLE V

| Methods         | YOLO | EKF | OSP | DBSCAN | KSP | Tracking |
|-----------------|------|-----|-----|--------|-----|----------|
| DYP-SLAM        | 12.44| 0.07| 17.93| 1.76   | 3.66| 42.7     |
| DYP-SLAM−       | 12.44| 0.07| /    | 1.76   | 3.66| 24.77    |

VI. Conclusion

In this paper, we propose a dynamic scene-oriented visual SLAM algorithm based on YOLOv5s and static probability. After missed detection compensation and keypoints clustering, the static probabilities of objects, keypoints and map points are calculated and updated as weights to participate in camera pose optimization. Extensive evaluation shows that our algorithm achieves the highest positioning accuracy in almost all low dynamic and high dynamic scenes, and has quite high real-time performance. In the next work, we intend to build a lightweight plane and object map containing only static environment for robot navigation and augmented reality.

REFERENCES

[1] M. R. U. Saputra, A. Markham, and N. Trigoni, “Visual slam and structure from motion in dynamic environments: A survey,” ACM Computing Surveys (CSUR), vol. 51, no. 2, pp. 1–36, 2018.

[2] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.

[3] C. Wang, T. Ji, T.-M. Nguyen, and L. Xie, “Correlation flow: robust optical flow using kernel cross-correlators,” in 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018, pp. 836–841.

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 12, pp. 2481–2495, 2017.

[5] R. Mur-Artal and J. D. Tardós, “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.

[6] 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.

[7] Y. Sun, M. Liu, and M. Q.-H. Meng, “Improving rgb-d slam in dynamic environments: A motion removal approach,” Robotics and Autonomous Systems, vol. 89, pp. 110–122, 2017.

[8] W. Dai, Y. Zhang, P. Li, Z. Fang, and S. Scherer, “Rgb-d slam in dynamic environments using point correlations,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020.

[9] 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.

[10] J. Bian, W.-Y. Lin, Y. Matsushita, S.-K. Yeung, T.-D. Nguyen, and M.-M. Cheng, “Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4181–4190.

[11] D.-H. Kim and J.-H. Kim, “Effective background model-based rgb-d dense visual odometry in a dynamic environment,” IEEE Transactions on Robotics, vol. 32, no. 6, pp. 1565–1573, 2016.

[12] T. Zhang, H. Zhang, Y. Li, Y. Nakamura, and L. Zhang, “Flowfusion: Dynamic dense rgb-d slam based on optical flow,” in 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020, pp. 7322–7328.

[13] F. Zhong, S. Wang, Z. Zhang, and Y. Wang, “Detect-slam: Making object detection and slam mutually beneficial,” in 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2018, pp. 1001–1010.

[14] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, “Ds-slam: A semantic visual slam towards dynamic environments,” in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018, pp. 1168–1174.

[15] B. Bescos, J. M. Fácil, 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.

[16] L. Xiao, J. Wang, X. Qiu, Z. Rong, and X. Zou, “Dynamic-slam: Semantic monocular visual localization and mapping based on deep learning in dynamic environment,” Robotics and Autonomous Systems, vol. 117, pp. 1–16, 2019.

[17] X. Yuan and S. Chen, “Sad-slam: A visual slam based on semantic and depth information,” in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020, pp. 4930–4935.

[18] A. Li, J. Wang, M. Xu, and Z. Chen, “Dp-slam: A visual slam with moving probability towards dynamic environments,” Information Sciences, vol. 556, pp. 128–142, 2021.

[19] Y. Fan, Q. Zhang, Y. Tang, S. Liu, and H. Han, “Blitz-slam: A semantic slam in dynamic environments,” Pattern Recognition, vol. 121, p. 10825, 2022.

[20] T. Ji, C. Wang, and L. Xie, “Towards real-time semantic rgb-d slam in dynamic environments,” arXiv preprint arXiv:2104.01316, 2021.

[21] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “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.