Received 26 October 2023, accepted 8 December 2023, date of publication 12 December 2023, date of current version 19 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3342080

RESEARCH ARTICLE

YG-SLAM: Enhancing Visual SLAM in Dynamic Environments With YOLOv8 and Geometric Constraints

GUOMING CHU[1], YAN PENG[1,2], XUHONG LUO[1], AND JING GONG[1]
[1]School of Automation and Information Engineering, Sichuan University of Science and Engineering, Yibin 644002, China
[2]Key Laboratory of Higher Education of Sichuan Province for Enterprise Informationalization and Internet of Things, Sichuan University of Science and Engineering, Yibin 644002, China

Corresponding author: Guoming Chu (321081104110@stu.suse.edu.cn)

This work was supported in part by the Opening Project of Key Laboratory of Higher Education of Sichuan Province for Enterprise Informationalization and Internet of Things under Grant 2021WYJ04.

ABSTRACT In dynamic environments, achieving accurate and robust Visual SLAM (Simultaneous Localization and Mapping) remains a significant challenge, particularly for applications in robotic navigation and autonomous driving. This study introduces YG-SLAM, an innovative approach that integrates YOLOv8 and geometric constraints within the ORB-SLAM2 framework to adapt effectively to dynamic scenarios.YOLOv8 is employed for instance segmentation and dynamic object detection, enriching the semantic information while extracting image feature points. Geometric constraints, including epipolar geometry algorithms and Lucas-Kanade optical flow methods, are utilized to filter out dynamic objects effectively.The tracking thread exclusively relies on static feature points for camera pose estimation, substantially improving the system's localization accuracy. Experimental results on the TUM dataset demonstrate that YG-SLAM significantly outperforms traditional ORB-SLAM2 in dynamic environments. Specifically, the Root Mean Square Error (RMSE) of the absolute trajectory errors reduced by 96.51% in comparison to ORB-SLAM2, and the RMSE of the relative pose errors decreased by 93.60% when compared to the performance of ORB-SLAM2.These notable reductions in errors demonstrate the promising performance enhancements of YG-SLAM over traditional ORB-SLAM2 in dynamic environments.

INDEX TERMS Dynamic environments, geometric constraints, instance segmentation, object detection, visual SLAM.

I. INTRODUCTION

With the widespread application of robots in areas such as social service, public safety, and disaster relief, Visual Simultaneous Localization and Mapping (Visual SLAM) has gained significant attention. This technology involves the synchronized construction of a structural map and self-localization of robots using visual sensors to perceive the surrounding environment [1], [2], [3], [4]. However, traditional SLAM algorithms based on geometric vision often suffer from significant pose estimation errors, especially in the presence of dynamic objects in the environment. Concurrently, the combination of deep learning and geometric constraints offers a promising approach to address this challenge.

While several Visual SLAM methods like LSD-SLAM [5], ORB-SLAM [6], [7], VINS-Mono [8], and DSO [9] have achieved remarkable real-time performance and accuracy, they are mainly effective in static environments. These methods face challenges when dynamic objects introduce inaccuracies in both mapping and localization [10], [11]. Recent advances in deep learning technologies have spurred innovative research aimed at overcoming these limitations. Specifically, semantic constraints have been employed to resolve Visual SLAM challenges in dynamic settings. The common approach is to leverage semantic information, derived either from object detection methods such as [12] and [13] or from semantic segmentation algorithms like [14],

The associate editor coordinating the review of this manuscript and approving it for publication was Yassine Maleh.

[15], [16], [17], [18], [19], [20], [21], [22], [23], and [24] as prior knowledge. This information is then amalgamated with geometric constraints to filter out dynamic objects from the environment. While promising, each of these methods comes with its own set of limitations. For instance, semantic segmentation offers detailed, pixel-level object masks but falls short in real-time performance. Efforts to improve its accuracy often lead to significant computational overheads [22]. Algorithms such as MID-Fusion [25] and [26] do offer solutions for the detection of dynamic objects, but their efficiency substantially declines when the dynamic objects are large in scale. Other SLAM systems aiming to incorporate dynamic object detection, such as Dynamic-SLAM [13] and MVO [27], often suffer from lengthy detection times.It becomes increasingly evident that the integration of deep learning methods and geometric constraints in SLAM can offer a robust solution for dynamic environments, reinforcing the significance of our study.

In response to the challenges dynamic objects impose on SLAM algorithms, this study introduces a cutting-edge front-end visual odometry method specifically engineered for the removal of dynamic objects. Our innovative approach seamlessly integrates YOLOv8's real-time object detection and image segmentation functionalities into the ORB-SLAM2 framework [7]. A newly incorporated instance segmentation thread capitalizes on YOLOv8's capabilities to extract rich semantic information from the scene. This dual-pronged strategy not only leverages state-of-the-art deep learning techniques for precise object motion characterization but also employs sparse optical flow to validate the dynamic states of these objects.

1) This study successfully integrated real-time object detection and image segmentation technologies from YOLOv8 into the ORB-SLAM2 framework, introducing a new instance segmentation thread. As a result, it achieved highly accurate and robust visual SLAM in dynamic environments.
2) This study proposed a two-step dynamic point detection algorithm, including pre-detection and epipolar constraint [28]. The algorithm utilizes feature point matching and fundamental matrix computation using LK optical flow [29] and the eight-point method. It then accurately identifies dynamic points through epipolar lines and threshold criteria.
3) This study further advanced the integration of semantic information into visual SLAM by proposing a novel algorithm for indoor map construction based on semantic information. Building indoor maps enriched with semantic details enhances the precision of mobile robot localization and navigation, augmenting the robot's understanding of the environment.
4) Experimental evidence validates that our method substantially outperforms existing state-of-the-art visual SLAM algorithms, particularly in terms of pose estimation accuracy within dynamic environments.

II. RELATED WORK

A. VSLAM ROBUSTNESS AGAINST DYNAMIC ENVIRONMENTS

Methods for addressing the challenges of dynamic scenes in existing visual SLAM can be broadly categorized into three classes, as outlined in [30].

1) GEOMETRIC CONSTRAINTS-BASED METHODS

In geometric methods, the assumption is that only static features conform to geometric constraints, while dynamic features do not. These methods often leverage the distinct characteristics of dynamic objects moving against a static backdrop. Li et al. [31] selected depth edge points for correspondence and formulated a static weighting technique to minimize the influence of dynamic points. Wang et al. [32] employed the fundamental matrix to identify inconsistencies in feature points and subsequently clustered the depth image; areas with outlier counts exceeding a certain threshold were flagged as containing moving objects. Cheng et al. [33] introduced a sparse motion removal model that relies on the similarities and differences between consecutive frames to detect dynamic regions. However, geometric methods typically require a predefined threshold to distinguish between dynamic and static feature points, which can lead to either over-recognition or under-recognition.Palazzolo et al. [34] introduced Refusion, a geometry-based method for robust dense indoor mapping in dynamic environments, utilizing geometric residuals and an empirical threshold to segment dynamic objects without reliance on semantic interpretation or object-specific detectors [35].Dai et al. use Delaunay triangulation and point correlations to separate static and dynamic elements in RGB-D SLAM.

2) OPTICAL FLOW-BASED METHODS

Optical flow estimates the pixel motion between two consecutive frames in luminance mode, often serving as a motion field to segment moving objects [36]. Scene flow, considered as the 3D extension of optical flow, along with 2D optical flow, provides descriptions of various moving objects in 3D point clouds and 2D images.Derome et al. [37], [38] utilized the residual between predicted and observed images from stereo cameras to calculate optical flow. By transforming the current frame to the previous one using estimated camera ego-motion, they detect moving objects in the residual field.

Bakkay et al. employed scene flow estimation and used a region-growing segmentation algorithm for dynamic object segmentation, enabling real-time 3D static background mapping using a Kinect sensor [39].FlowFusion [40] employs optical flow residuals and geometric information for dynamic object segmentation and static background reconstruction in RGB-D SLAM. It utilizes PWC-Net [41] for optical flow estimation and calculates 2D scene flow for dynamic segmentation. Despite its accuracy, its real-time performance could be improved due to computational demands

3) DEEP LEARNING-BASED METHODS

In recent years, deep learning-based image semantic segmentation and object detection methods have made significant advancements in both efficiency and accuracy. Many researchers have attempted to address dynamic SLAM challenges by using semantic labeling or object detection as preprocessing steps to remove potential dynamic objects.DS-SLAM [14]uses SegNet [42] for the purpose of semantic segmentation and employs a moving consistency check to eliminate dynamic elements from the scene, thereby generating a comprehensive semantic octo-tree map.In Detect-SLAM [12], the system identifies objects within keyframes and continuously propagates the motion probability of keypoints to promptly mitigate the impact of dynamic objects on the SLAM process.MaskFusion [43] incorporates Mask R-CNN [44] to achieve instance-level segmentation with semantic labeling, enabling the handling of moving objects.DynaSLAM [15] employs Mask R-CNN [44] and multiview geometric models to identify and process moving objects, offering the added benefit of restoring backgrounds that are obscured by dynamic elements.Vincent et al. [19] conduct semantic segmentation of object instances within the image and employ an Extended Kalman Filter (EKF) to recognize, track, and eliminate dynamic objects from the scene.DP-SLAM [20] integrates the outcomes of geometric constraints and semantic segmentation, tracking dynamic keypoints within a Bayesian probability estimation framework.Ji et al. [21] conduct semantic segmentation exclusively on keyframes. They then cluster the depth map and utilize this information in conjunction with the re-projection error to detect and eliminate both known and unknown dynamic objects.Blitz-SLAM [22] reconstructs the BlitzNet [45] mask by incorporating depth information and employs epipolar constraints to differentiate between static and dynamic matching points within potential dynamic regions.DGS-SLAM [46] presents a dynamic RGBD SLAM system that integrates geometric and semantic information, employing a dynamic object detection module based on a multi-residual model and a camera pose tracking strategy with feature point classification to extract potential moving objects.

Ran et al. [47] proposed a novel RGB-D-inertial dynamic SLAM method that enables accurate localization even when a significant portion of the camera's view is obstructed by multiple dynamic objects for an extended period.SVD-SLAM [48] has incorporated an improved YOLACT++ lightweight instance segmentation network and calculates the motion probability of each potential moving object based on the camera pose and polar constraints.DIG-SLAM [49] utilizes YOLOv7 instance segmentation to extract object contours, while optimizing line features through line segment detection and K-means clustering. It employs motion consistency checks to identify dynamic areas and removes points and line features within them. DYS-SLAM [50].

Current literature mainly leans towards combining geometric constraints and deep learning for Visual SLAM in dynamic settings. These methods, while effective in removing dynamic objects, often compromise between real-time performance and localization accuracy. To address these limitations, our study introduces a streamlined approach that tightly integrates YOLOv8's real-time object detection with geometric techniques like epipolar constraints and LK optical flow. This innovation not only reduces computational load but also enhances accuracy, offering a more practical and stable SLAM system for dynamic environments.

III. PROPOSED METHOD

A. FRAMEWORK OF SYSTEM

In our proposed system, we seamlessly integrate the robust SLAM capabilities of ORB-SLAM2 with a parallel-running instance segmentation thread. This architecture is specifically designed for dynamic environments, focusing on camera pose estimation and point cloud map construction. The system comprises two main threads: the tracking thread and the instance segmentation thread. The tracking thread takes RGB and depth images as input and is responsible for estimating the camera pose, deciding whether to insert keyframes. It employs a series of advanced techniques, including ORB feature extraction and matching, optical flow calculation of the sparse feature set using the Lucas-Kanade [29] method, and fundamental matrix computation for the filtered optical flow points. These points are then validated through epipolar constraints, and those not meeting the constraints are identified as dynamic points. Concurrently, the instance segmentation thread employs YOLOv8 to perform instance segmentation on RGB images, identifying potential dynamic objects and then collaborating with a motion consistency check algorithm to eliminate them. These two threads work in tandem to construct a semantically rich map, where the instance segmentation thread enhances the expressiveness of keyframes, and the tracking thread improves pose estimation accuracy through motion consistency checks.The overall framework of the proposed method is shown in Fig.1.

B. INSTANCE SEGMENTATION BASED ON YOLOv8

The YOLOv8 model structure is designed to achieve efficient and accurate instance segmentation. Its core idea is to divide the image into grids and predict the category, location, and segmentation mask of the target in each grid. The model consists of a Backbone network and a Head. The Backbone network is responsible for extracting the features of the image, while the Head is responsible for predicting the category, location, and segmentation mask of the target. The Backbone network uses Darknet as its basis, which is a lightweight convolutional neural network structure. Darknet consists of multiple convolutional layers

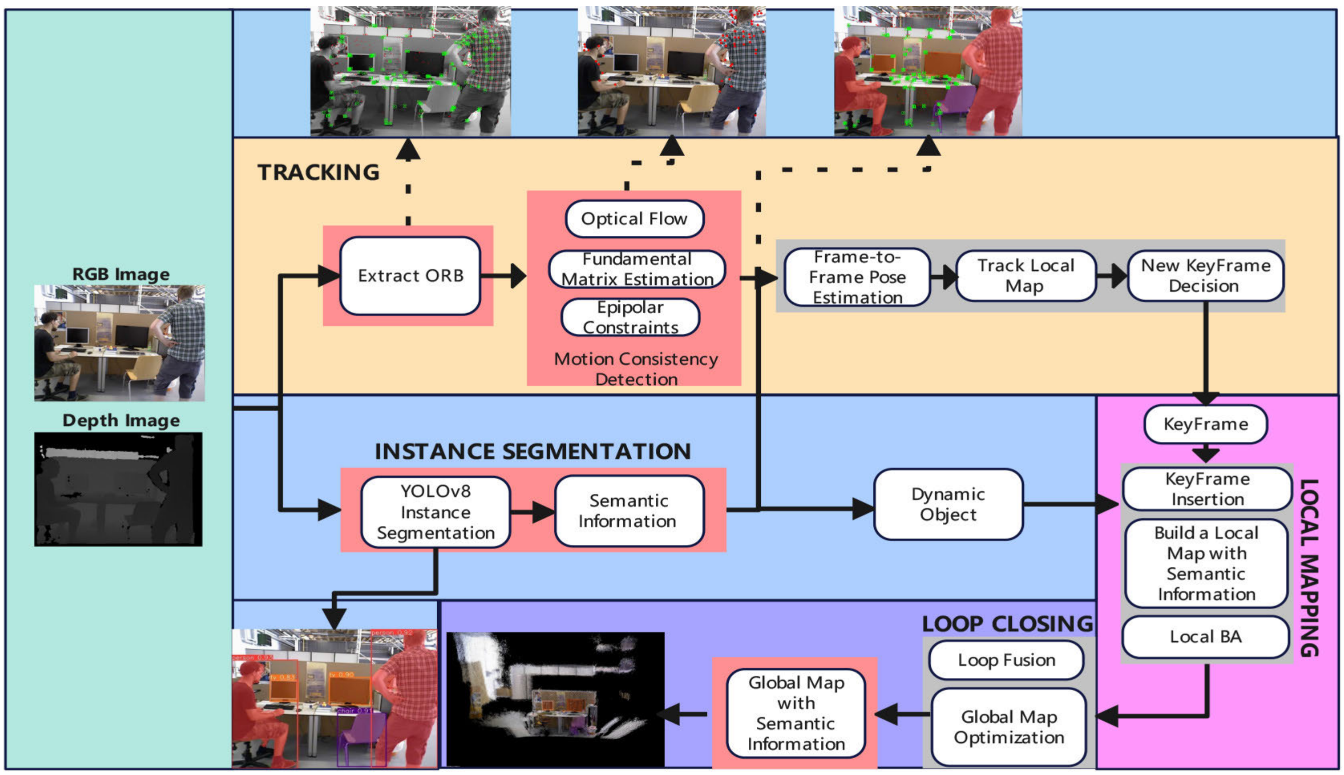


FIGURE 1. The framework of our approach consists of four threads. The tracking thread and instance segmentation thread based on YOLOv8 run in parallel, while local mapping and loop closure are the same as ORB-SLAM2. After removing dynamic points, a point cloud map with semantic information is created.

and pooling layers, and by stacking these layers, it achieves effective feature extraction. YOLOv8 improves upon Darknet by introducing residual connections and skip connections, thereby enhancing the network's feature expression ability and perception range.

To improve the accuracy of the model, YOLOv8 also introduces an attention mechanism and multi-scale feature fusion. The attention mechanism adaptively adjusts the weights in the feature map, making the model focus more on important feature regions. Multi-scale feature fusion enhances the model's ability to perceive targets of different scales by fusing feature maps at different levels. The training process of the YOLOv8 model structure consists of two stages: first, target detection is performed to generate candidate object boxes by predicting the category and location of the target. Then, in the instance segmentation stage, the model uses these candidate object boxes to generate segmentation masks and optimizes these masks to obtain more accurate segmentation results, as shown in Fig. 2.

The design of the YOLOv8 instance segmentation model structure takes into account key issues such as feature extraction, target prediction, and segmentation optimization. By introducing technical means like residual connections, skip connections, attention mechanisms, and multi-scale feature fusion, the model's accuracy and robustness are enhanced. YOLOv8 has broad application prospects in the field of instance segmentation and will provide strong support for the research and application of instance segmentation tasks.

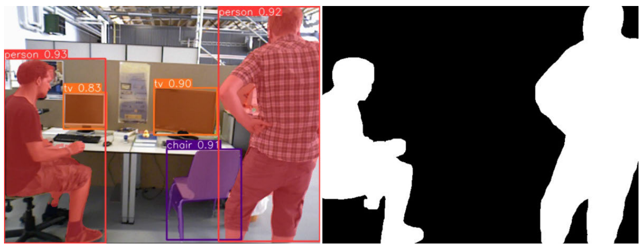


FIGURE 2. Target detection and instance segmentation based on YOLOv8.

C. MOTION OBJECT DETECTION BASED ON OPTICAL FLOW

Optical flow serves as a mechanism to capture the motion dynamics between two consecutive frames by correlating individual pixels. Unlike methods that rely on feature descriptor computation and feature matching for tracking pixels, optical flow boasts superior real-time performance. In this study, we employ the Lucas-Kanade optical flow technique to monitor feature points and compute the fundamental matrix, thereby facilitating the identification of dynamic feature

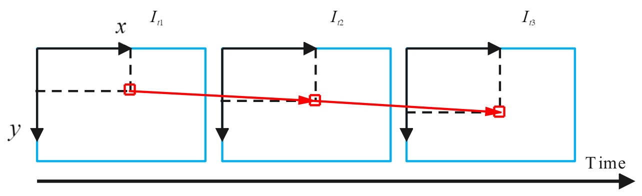


FIGURE 3. Lucas-Kanade optical flow diagram.

points. A schematic representation of the Lucas-Kanade optical flow approach is illustrated in Fig.3.

In the Lucas-Kanade (LK) [29] optical flow, we consider images from the camera to be time-varying. An image can be considered as a function of time: $\boldsymbol{I}(t)$. The gray level of a pixel at coordinates (x, y) at time t can be represented as $\boldsymbol{I}(x, y, t)$. Assuming that the gray level of a point in space projected onto all images is constant, then at time $t + dt$, if the pixel moves to $(x+dx, y+dy)$. Based on the brightness constancy assumption, we have:

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

Assuming that the motion between two image frames is relatively small, we perform a Taylor expansion on the left-hand side, retaining the first-order term:

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

According to the brightness constancy assumption:

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

Dividing both sides by dt, we get:

$$\frac{\partial \boldsymbol{I}}{\partial x}\frac{dx}{dt} + \frac{\partial \boldsymbol{I}}{\partial y}\frac{dy}{dt} = -\frac{\partial \boldsymbol{I}}{\partial t} \tag{4}$$

where $u = dx/dt$ represents the velocity component of the pixel along the x-axis, and $v = dy/dt$ represents the velocity component along the y-axis. Let $\boldsymbol{I}_x$ and $\boldsymbol{I}_y$ be the image gradients in the x and y directions, respectively. The equation can be rewritten as:

$$[\boldsymbol{I}_x \ \boldsymbol{I}_y]\begin{bmatrix} u \\ v \end{bmatrix} = -\boldsymbol{I}_t \tag{5}$$

To compute u and v, the Lucas-Kanade optical flow algorithm assumes that pixels within an image block of size $w \times w$ have the same motion, resulting in w^2 equations:

$$[\boldsymbol{I}_x \ \boldsymbol{I}_y]_k\begin{bmatrix} u \\ v \end{bmatrix} = -\boldsymbol{I}_{t_k}, \quad k = 1, \ldots, w^2 \tag{6}$$

Let $\boldsymbol{A}$ and b be defined as:

$$\boldsymbol{A} = \begin{bmatrix} [\boldsymbol{I}_x, \boldsymbol{I}_y]_1 \\ \vdots \\ [\boldsymbol{I}_x, \boldsymbol{I}_y]_k \end{bmatrix}, b = \begin{bmatrix} \boldsymbol{I}_{t1} \\ \vdots \\ \boldsymbol{I}_{tk} \end{bmatrix}$$

The entire equation becomes:

$$\boldsymbol{A}\begin{bmatrix} u \\ v \end{bmatrix} = -b \tag{7}$$

Using the traditional least squares solution, we get:

$$\begin{bmatrix} u \\ v \end{bmatrix}^* = -(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T b \tag{8}$$

Finally, this gives us the motion velocity u; v of pixels between images. When t takes discrete moments instead of continuous time, we can estimate the position of a block of pixels in several images. Since the pixel gradient is only locally valid, iterating this equation several times can obtain the motion of pixels in the image, thereby achieving pixel tracking.

D. MOTION CONSISTENCY DETECTION ALGORITHM

This paper addresses the issue of low accuracy in dynamic feature point recognition in high-dynamic scenes based on the motion consistency checking method proposed by Yu et al. [14]. Building upon their approach, we introduce a novel motion consistency checking method.

Yu et al. proposed a motion consistency checking method that primarily relies on geometric information for dynamic feature point extraction, with the following concise steps:

1) Employing optical flow pyramids to obtain matching feature points between consecutive frames.
2) Estimating the fundamental matrix using the RANSAC (Random Sample Consensus) [51] method to mitigate the influence of outliers and noise.
3) Utilizing epipolar geometry constraints to assess the motion properties of feature points, where dynamic feature points typically fail to satisfy epipolar geometry constraints.

In scenes with few dynamic objects, RANSAC effectively estimates model parameters and minimizes the impact of a limited number of dynamic feature points. However, in highly dynamic scenes with numerous dynamic points, this approach is less suitable. To address this, we propose an improved motion consistency detection method that combines image priors and optical flow, building upon the [14] method. The algorithm has two main steps: First, we accurately estimate the fundamental matrix **F** based on the motion characteristics of feature points in the previous frame's image. Then, we use geometric constraints to identify dynamic points. See Algorithm 1 for an overview of the workflow.Specific steps are as follows:

1) In this process, we did not employ the Harris corner detection method as used in [14]. Instead, we directly utilized the ORB features extracted by the ORB-SLAM2 system. This approach eliminates the need for additional computation of Harris corners. Subsequently, we applied the pyramid-based Lucas-Kanade optical flow algorithm to track all ORB feature points extracted from the previous frame, including both

dynamic and static feature points, in order to obtain feature correspondences between the two frames.

2) In the process of selecting reliable feature correspondences, we rely on the feature points from the previous frame, which have been categorized into two groups: dynamic feature points and static feature points. Therefore, in this paper, we exclusively utilize feature correspondences involving static feature points from the previous frame to compute the fundamental matrix. Additionally, we perform filtering by discarding feature correspondences that are either too close to the image edges or have pixel differences within the 3×3 image block center exceeding a predefined threshold.
3) Matching and estimating the fundamental matrix **F** are performed using reliable static feature points from the previous frame. In this research, we employ the RANSAC method to compute the fundamental matrix **F** with the aim of minimizing the interference caused by outliers.
4) After obtaining the fundamental matrix between adjacent frames, dynamic point extraction is performed based on epipolar line constraints.

1) FUNDAMENTAL MATRIX ESTIMATION BASED ON IMAGE PRIOR

After obtaining the feature matching relationships between adjacent frames using the Lucas-Kanade optical flow method, it is necessary to estimate the fundamental matrix between the two frames for subsequent determination of the motion attributes of feature points based on epipolar geometry constraints. Due to the presence of numerous dynamic points in the scene, directly computing the fundamental matrix between the two camera frames based on the feature matching obtained from the optical flow method can result in poor accuracy. This, in turn, leads to errors in filtering dynamic feature points based on epipolar geometry constraints, ultimately significantly affecting the localization accuracy of the ORB-SLAM2 algorithm. Therefore, in this paper, we leverage the motion attribute information of feature points in the previous frame to compute the fundamental matrix solely based on the matching of static feature points from the previous frame to the current frame.

The Fundamental Matrix F with elements f_i is given by

$$F = \begin{pmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{pmatrix}. \tag{9}$$

It has 5 degrees of freedom and can be computed from at least 5 feature point pairs. In this paper, we use the eight-point method, resulting in a linear algebraic problem.

Given a matching pair p_1 and p_2 with homogeneous coordinates $P_1 = [u_1, v_1, 1]$ and $P_2 = [u_2, v_2, 1]$, the epipolar constraint is

$$P_2^T F P_1 = 0. \tag{10}$$

Algorithm 1 Modified ProcessMovingObject Algorithm

```
Require: Gray Image (imgray), Previous Gray Image (imGrayPre)
Ensure: Filtered Key Points ($F2\_prepoint$, $F2\_nextpoint$), Anomalies ($T\_M$)
  Initialize and clear old data: $F\_prepoint$, $F\_nextpoint$, $F2\_prepoint$, $F2\_nextpoint$, $T\_M$;
  Extract ORB features from imGrayPre and store them in $F\_prepoint$;
  Use Pyramidal Lucas-Kanade method to track ORB features in imgray, resulting in nextpoint;
  for each prepoint $i$ do
    if state[i] != 0 then
      Check if the points are within the image boundary;
      if points are outside the boundary or other criteria not met then
        $state[i] = 0$;
      else
        Add prepoint[i] to $F\_prepoint$;
        Add nextpoint[i] to $F\_nextpoint$;
      end if
    end if
  end for
  Use RANSAC to compute the Fundamental Matrix $F$ only based on static feature points from imGrayPre and their matches in $F\_nextpoint$;
  for each point in $F\_prepoint$ and $F\_nextpoint$ do
    Calculate distance to the epipolar line;
    if distance $<= \tau$ then
      Add point to $F2\_prepoint$ and $F2\_nextpoint$;
    end if
  end for
  for each point in nextpoint do
    if state[i] != 0 then
      Calculate distance to the epipolar line;
      if distance $> limit_\tau$ then
        Add point to $T\_M$;
      end if
    end if
  end for
```

Rewriting the constraint yields

$$(u_1, v_1, 1)F \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} = 0. \tag{11}$$

Let

$$f = (f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9)^T \tag{12}$$

then

$$(u_1u_2, u_1v_2, u_1, v_1u_2, v_1v_2, v_1, u_2, v_2, 1)f = 0. \tag{13}$$

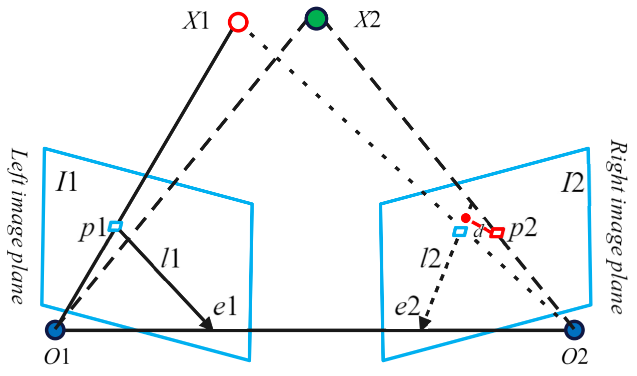


FIGURE 4. **Epipolar constraint relationship between two images.**

With 8 point pairs, we formulate a system of equations to solve for F. To account for outliers, we employ the RANSAC algorithm to refine the Fundamental Matrix computation.

2) DYNAMIC FEATURE POINT DISCRIMINATION BASED ON EPIPOLAR CONSTRAINT

The feature points from the previous frame, when mapped to the current frame area through the fundamental matrix, form the epipolar line. The epipolar constraint can be used to inspect the static and dynamic attributes of a point. When there are dynamic objects in the scene, the spatial points on the dynamic objects corresponding to two pixel coordinates and the fundamental matrix F will not satisfy the epipolar constraint relationship. The pixel point and its corresponding epipolar line will have a distance deviation, as shown in Fig.4.

Let the feature matching points in the previous frame image and the current frame image be represented as $p1 = [u1, v1]$ and $p2 = [u2, v2]$ respectively, where u, v are the image pixel coordinate values. The homogeneous coordinates of the feature point can be expressed as $P1 = [u1, v1, 1], P2 = [u2, v2, 1]$. If the fundamental matrix is represented as F, then the epipolar constraint is expressed as:

$$P_2^T F P_1 = 0 \tag{14}$$

For the epipolar line l of the feature $P1$ in the current frame,

$$l = FP_1 = [A, B, C]^T \tag{15}$$

Then, for the point $P2$ in the current frame, the distance to the corresponding epipolar line is computed as:

$$D = \frac{|P_2^T F P_1|}{\sqrt{A^2 + B^2}} \tag{16}$$

For each feature point in the current frame, if its distance D to the corresponding epipolar line is greater than a threshold τ, it is added to the set of potential dynamic feature points.

IV. EXPERIMENTAL RESULTS

To validate the localization accuracy of our proposed visual SLAM algorithm that removes dynamic objects, tests were conducted using the TUM RGB-D dataset [52]. The TUM RGB-D dataset is comprised of 39 sequences, captured with a Microsoft Kinect sensor in a variety of indoor settings at a full frame rate of 30Hz. A detailed description of the TUM dataset sequences is presented in Table 1. This dataset offers a range of typical SLAM scenarios, such as Handheld SLAM and Robot SLAM, making it particularly suited for our study focused on indoor dynamic motion scenes. For the purpose of these tests, we specifically utilized the dynamic object datasets available within the TUM RGB-D dataset.

Based on the speed and nature of characters in the scene, datasets were classified into two categories: low dynamic (sitting) and high dynamic (walking). In the low dynamic or 'sitting' (s) sequences, two people are seen sitting in front of a desk, engaged in speaking and gesticulating, thereby exhibiting a low degree of motion. On the other hand, the high dynamic or 'walking' (w) sequences feature two people walking both in the background and the foreground, later sitting down at the desk, making these sequences highly dynamic and challenging for standard SLAM systems. Each category consists of several typical camera motion patterns such as halfsphere, r-p-y (roll-pitch-yaw), static, and x-y-z. In the figures presented, the dataset's name adopts a specific format: "Dataset Name + Algorithm Type." High dynamic walking scenarios are abbreviated as "w," while low dynamic sitting scenarios are abbreviated as "s." For instance, the sequence fr3_w_h refers to a walking scenario with halfsphere camera motion in rgbd_datase_freiburg3.

A. ERROR METRICS

For quantitative evaluation of the algorithm's performance, the metrics Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) [52] are utilized. ATE assesses the global consistency of the trajectory, while RPE focuses on the algorithm's translational and rotational drift over time.Here we define the predicted trajectory and the true trajectory as $P_{1:n}$ and $Q_{1:n}$ respectively. Through Singular Value Decomposition, a transformation matrix T aligns the predicted trajectory with the true trajectory, and pose error is then computed. If $F_i := Q_i^{-1} T P_i$, where Q_i and P_i represent the true and predicted trajectories of the i-th keyframe respectively, ATE is defined as:

$$\mathrm{RMSE}(F_{1:n}) := \sqrt{\left(\frac{1}{n}\sum_{i=1}^{n} \|\mathrm{trans}(F_i)\|^2\right)} \tag{17}$$

Relative Pose Error (RPE) measures the local accuracy of the estimated trajectory over a fixed time interval Δt, including both translational and rotational errors. The relative pose error at the i-th point, F_i, is:

$$F_i := \left(Q_i^{-1} Q_{i+\Delta t}\right)^{-1} \left(P_i^{-1} P_{i+\Delta t}\right) \tag{18}$$

TABLE 1. **TUM dataset sequence descriptions.**

Sequence Name	Duration(s)	Ground-truth Length(m)	Avg. Trans. Velocity(m/s)	Avg. Angular Velocity(deg/s)	Description
fr3_s_s	23.63	0.259	0.011	1.699	The camera remains stationary.
fr3_w_h	35.81	7.686	0.221	18.267	The camera moves on a small hemisphere with a diameter of one meter.
fr3_w_r	30.61	2.698	0.091	20.903	The camera rotates around the main axes (roll-pitch-yaw) at the same location.
fr3_w_s	24.83	0.282	0.012	1.388	The camera remains stationary.
fr3_w_x	28.83	5.791	0.208	5.490	The camera moves along three directions (xyz) while maintaining the same orientation.

The Root Mean Square Error (RMSE) over all time points serves as the RPE:

$$\mathrm{RMSE}(E_{1:n}, \Delta t) := \sqrt{\left(\frac{1}{m}\sum_{i=1}^{m} \|\mathrm{trans}(F_i)\|^2\right)} \tag{19}$$

where trans(F_i) denotes the translational elements within the relative pose error F_i.

The algorithm was developed using C++ on a system running Ubuntu 20.04. Hardware specifications include 32GB of RAM, an Intel i7-11800H CPU with a clock speed of 2.3GHz, and a GeForce GTX 3060 graphics card.

B. DYNAMIC FEATURE POINT REMOVAL

Fig.5 presents a comparison between the motion consistency check method proposed by Yu et al. [14] and the improved method introduced in this paper. In the figure, red points indicate dynamic feature points detected by the algorithm, while green points represent static feature points. From left to right, the figure sequentially displays the tracking results for frames 44, 68, 216, 499, 663, and 759 from fr3_walking_xyz dataset. The top row shows the tracking performance of the traditional motion consistency check algorithm, the middle row showcases the efficacy of the improved method proposed in this paper, and the bottom row illustrates the comprehensive tracking results achieved by combining the improved algorithm with the YOVOv8 semantic segmentation network. Compared to the method by Yu et al. [14], our improved method more accurately detects dynamic feature points on human bodies and has a lower false detection rate for static feature points in the background. This comparison validates the effectiveness of the method proposed in this paper. By integrating semantic segmentation and LK optical flow, we can not only identify dynamic humans but also more accurately remove dynamic points, as shown in the bottom row of Fig. 5.

C. EVALUATING TUM RGB-D DATASET PERFORMANCE

In this section, we benchmark our proposed YG-SLAM algorithm against ORB-SLAM2 [7] and other state-of-the-art dynamic scene SLAM algorithms, including Dyna-SLAM [15], DS-SLAM [14], and Blitz-SLAM [22]. All these algorithms, like ours, are built upon the enhanced ORB-SLAM2 framework.

Quantitative comparison results across different sequences and algorithms are presented in Table 2, 3, and 4. The best performance on each evaluation metric is highlighted in bold, and the second-best results are underlined. Data for Dyna-SLAM, DS-SLAM,and Blitz-SLAM are sourced from their respective publications, and a forward slash indicates that the data was not provided in the original papers. In this paper, we focus on RMSE (Root Mean Square Error) and S.D. (Standard Deviation) as they are more indicative of the system's robustness and stability. RMSE measures the deviation between the estimated and true values, while S.D. gauges the dispersion of the estimated camera trajectory. We also showcase the improvement of YG-SLAM over the original ORB-SLAM2, calculated as:

$$\eta = \frac{\alpha - \beta}{\alpha} \times 100\% \tag{20}$$

Here, η represents the improvement, α is the value for ORB-SLAM2, and β is the value for YG-SLAM.

From Tables 2 to 4, it is evident that YG-SLAM significantly outperforms ORB-SLAM2 in terms of ATE (Absolute Trajectory Error) and RPE (Relative Pose Error), achieving a reduction in overall localization error ranging from 89% to 98%.Specifically, for the ATE values presented in Table 2, in the high-dynamic sequence fr3_walking, the improvement in RMSE and S.D. can reach up to 98.09% and 98.13%, with average improvements of 96.51% and 94.46%, respectively. In the low-dynamic sequence fr3_sitting_static, the improvement is less pronounced, at 29.79% and 31.19%, due to the already competent performance of ORB-SLAM2 in low-dynamic scenarios, leaving limited room for improvement.

Tables 3 and 4 extend the quantitative analysis to include translational and rotational drift in RPE. In this context, it's worth noting the significance of the time interval parameter Δ_t. For multi-frame tracking systems like YG-SLAM, a larger Δ_t is often more effective. As an example, a Δ_t value of 30 allows us to assess drift on a

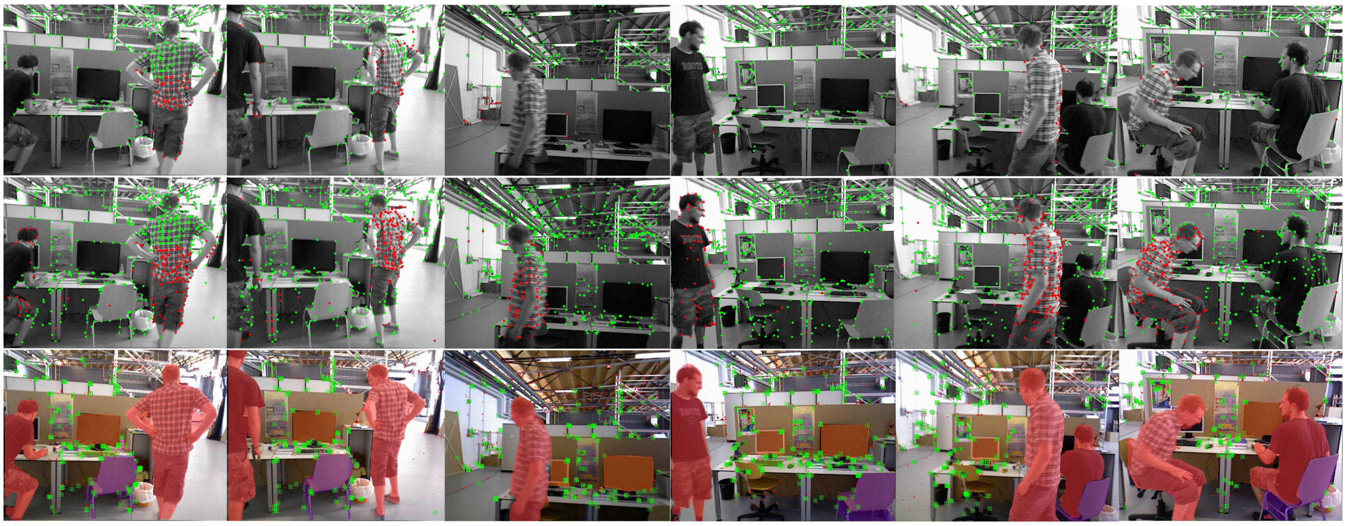


FIGURE 5. Displays the tracking comparison results of different algorithms. From left to right are the results of frames 44, 68, 216, 499, 663 and 759.The top row presents the tracking results obtained using the traditional motion consistency check algorithm. The middle row showcases the performance of the improved motion consistency check algorithm proposed in this paper. The bottom row represents the comprehensive tracking results achieved by combining this improved algorithm with the YOVOv8 semantic segmentation network.

TABLE 2. Analysis of absolute trajectory error (ATE[m]) metric.

Sequences	ORB-SLAM2		Dyna-SLAM		DS-SLAM		Blitz-SLAM		YG-SLAM		Error reduction rate against ORB-SLAM2	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
fr3_w_x	0.7842	0.4005	0.0164	0.0086	0.0247	0.0161	0.0153	0.0078	**0.0150**	**0.0075**	98.09%	98.13%
fr3_w_s	0.3872	0.1636	**0.0068**	**0.0032**	0.0081	0.0036	0.0102	0.0052	0.0079	0.0034	97.96%	97.92%
fr3_w_r	0.7842	0.4005	0.0354	0.0190	0.4442	0.2350	0.0356	0.0220	**0.0350**	**0.0188**	95.54%	95.13%
fr3_w_h	0.4667	0.2601	0.0296	0.0157	0.0303	0.0159	**0.0256**	**0.0126**	0.0259	0.0143	94.45%	94.50%
fr3_s_s	0.0087	0.0042	/	/	0.0065	0.0033	/	/	**0.0061**	**0.0029**	29.79%	31.19%

TABLE 3. Analysis of translational drift (RPE[m]) metric.

Sequences	ORB-SLAM2		Dyna-SLAM		DS-SLAM		Blitz-SLAM		YG-SLAM		Error reduction rate against ORB-SLAM2	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
fr3_w_x	0.3944	0.2964	0.0217	0.0119	0.0333	0.0229	0.0197	0.0096	**0.0115**	**0.0067**	97.09%	97.75%
fr3_w_s	0.2349	0.2151	0.0089	0.0044	0.0102	0.0048	0.0129	0.0069	**0.0077**	**0.0043**	96.71%	98.00%
fr3_w_r	0.4582	0.3447	0.0448	0.0262	0.1503	0.1168	0.0473	0.0283	**0.0312**	**0.0242**	93.20%	92.97%
fr3_w_h	0.3480	0.2859	0.0149	0.0149	0.0297	0.0152	**0.0253**	**0.0123**	0.0264	0.0130	92.43%	95.46%
fr3_s_s	0.0090	0.0043	/	/	0.0078	0.0038	/	/	**0.0049**	**0.0024**	45.11%	43.26%

TABLE 4. Analysis of rotational drift (RPE[deg]) metric.

Sequences	ORB-SLAM2		Dyna-SLAM		DS-SLAM		Blitz-SLAM		YG-SLAM		Error reduction rate against ORB-SLAM2	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
fr3_w_x	7.7846	5.8335	0.6284	0.3848	0.8266	0.5826	0.6132	0.3348	**0.4779**	**0.1730**	93.86%	97.03%
fr3_w_s	4.1856	3.8077	0.2612	0.1259	0.2690	0.1182	0.3038	0.1437	**0.2362**	**0.0966**	94.36%	97.46%
fr3_w_r	8.8923	6.6658	0.9894	0.5701	3.0042	2.3065	1.0841	0.6668	**0.7634**	**0.3095**	91.42%	95.36%
fr3_w_h	7.2138	5.8299	0.7842	0.4012	0.8142	0.4101	0.7879	0.3751	**0.7395**	**0.3998**	89.75%	93.14%
fr3_s_s	0.2850	0.1241	/	/	0.2735	0.1215	/	/	**0.2346**	**0.0920**	17.70%	25.86%

per-second basis in sequences captured at a rate of 30Hz [52]. The relative trajectory errors observed are in line with the absolute errors. In high-dynamic sequences, our method achieves an average RMSE and average S.D. reduction of 94.86% and 96.04%, respectively, for translational drift. For rotational drift, the average RMSE and average S.D. are

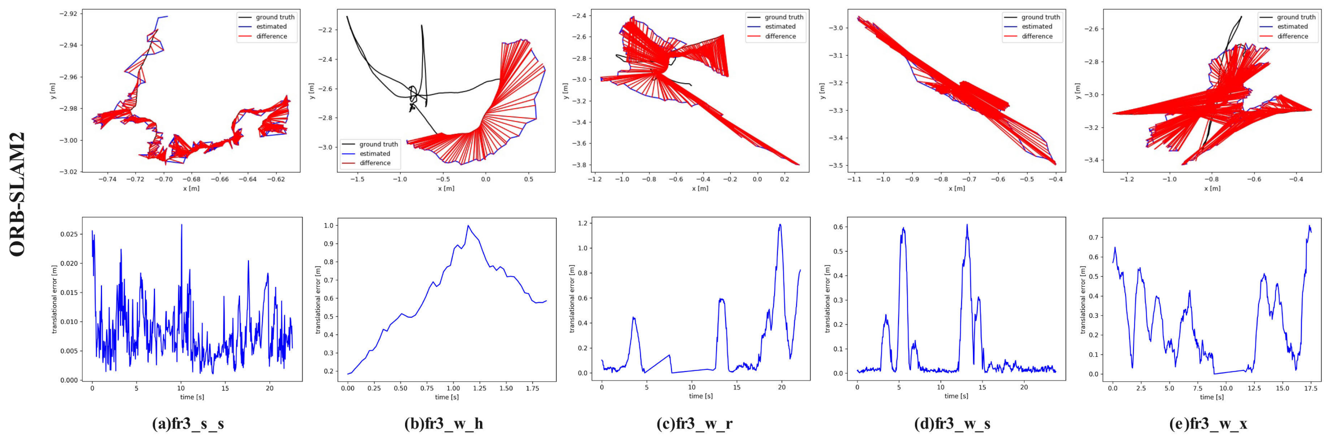


FIGURE 6. ATE and RPE from ORB-SLAM2.The diagram in the first row represents the absolute trajectory error (ATE), while the diagram in the second row illustrates the relative pose error (RPE).

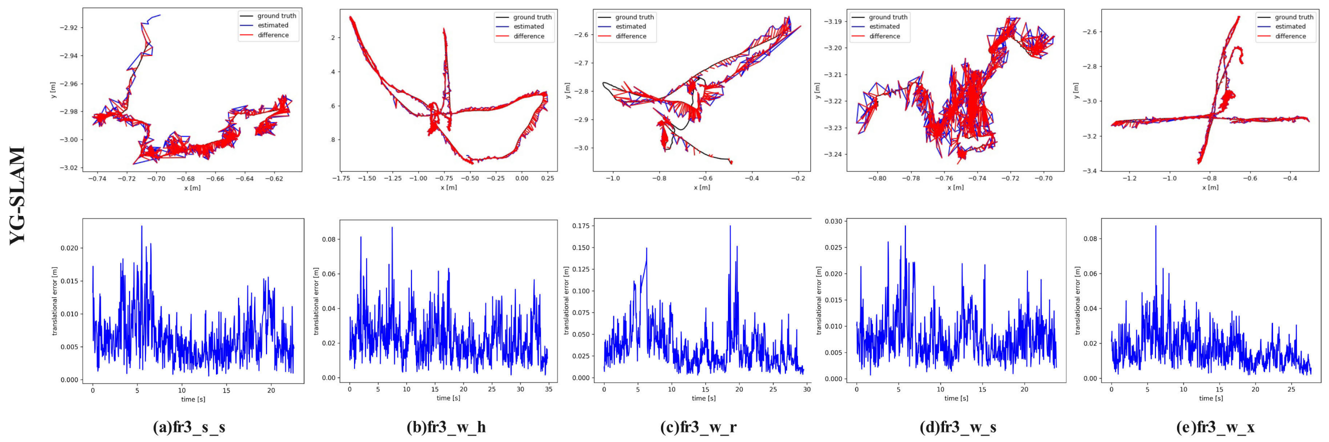


FIGURE 7. ATE and RPE from ORB-SLAM2.The diagram in the first row represents the absolute trajectory error (ATE), while the diagram in the second row illustrates the relative pose error (RPE).

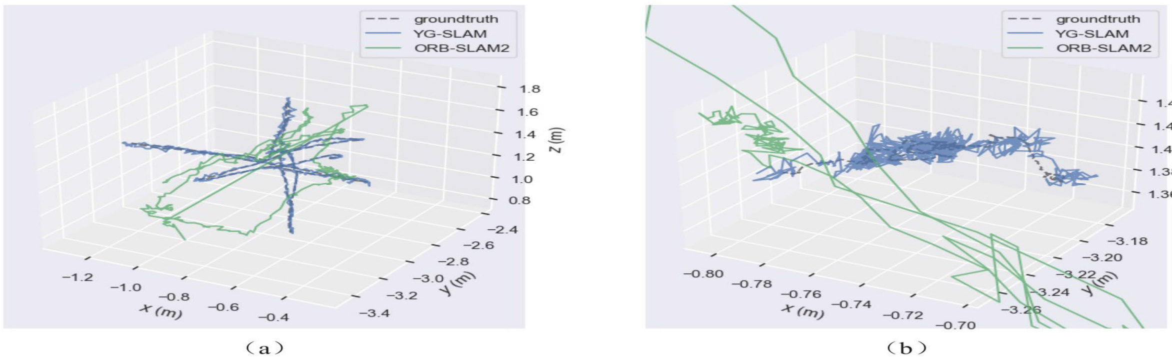
reduced by 92.35% and 95.75%, respectively. Conversely, in the low-dynamic fr3_sitting_static sequence, where static features are predominant, the robust camera pose estimation by ORB-SLAM2 leaves little room for further improvement. In this case, our method manages to reduce the RMSE for translational and rotational drift by 45.11% and 17.70%, respectively, and the S.D. for translational and rotational drift are reduced by 43.26% and 25.86%, respectively.

Fig.6 depicts the ATE and RPE calculated based on the original ORB-SLAM2 system. Fig.7 depicts the ATE and RPE calculated based on our proposed YG-SLAM system. The relative translational errors are visually represented by the length of the red lines. Qualitatively, it is evident that YG-SLAM estimates a camera trajectory that more closely aligns with the ground truth, significantly reducing errors. Our method keeps both ATE and RPE at a much lower level.Fig.8 shows the 3D trajectory of the camera motion estimated by YG-SLAM.

In summary, both qualitative and quantitative results are in agreement, demonstrating that our algorithm significantly enhances the localization accuracy of ORB-SLAM2 in both high-dynamic and low-dynamic environments.

D. TIMING ANALYSIS

In practical applications, real-time performance is a critical factor for evaluating the quality of SLAM (Simultaneous Localization and Mapping) systems. In this section, tests were conducted on the TUM dataset to assess the tracking time of each system. Table 5 displays the time costs for different SLAM systems. YG-SLAM has an average tracking time of approximately 42 ms per frame, which is significantly better than DynaSLAM. YG-SLAM utilizes the yolov8n-seg model of YOLOv8 for instance segmentation, with an average instance segmentation time per frame of 21.10 ms, greatly reducing the computational burden.



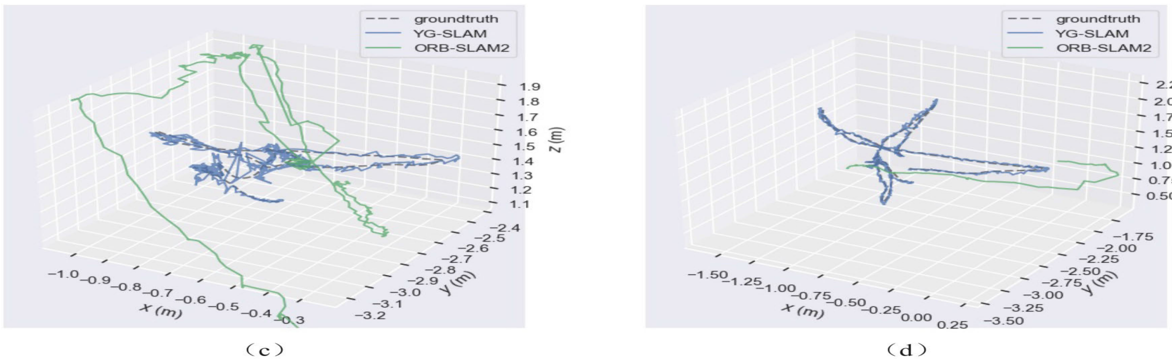


FIGURE 8. Comparison of experimental trajectories and real trajectories of YG-SLAM on different data sets. (a) fr3_w_x (b) fr3_w_s (c) fr3_w_r (d) fr3_w_h.

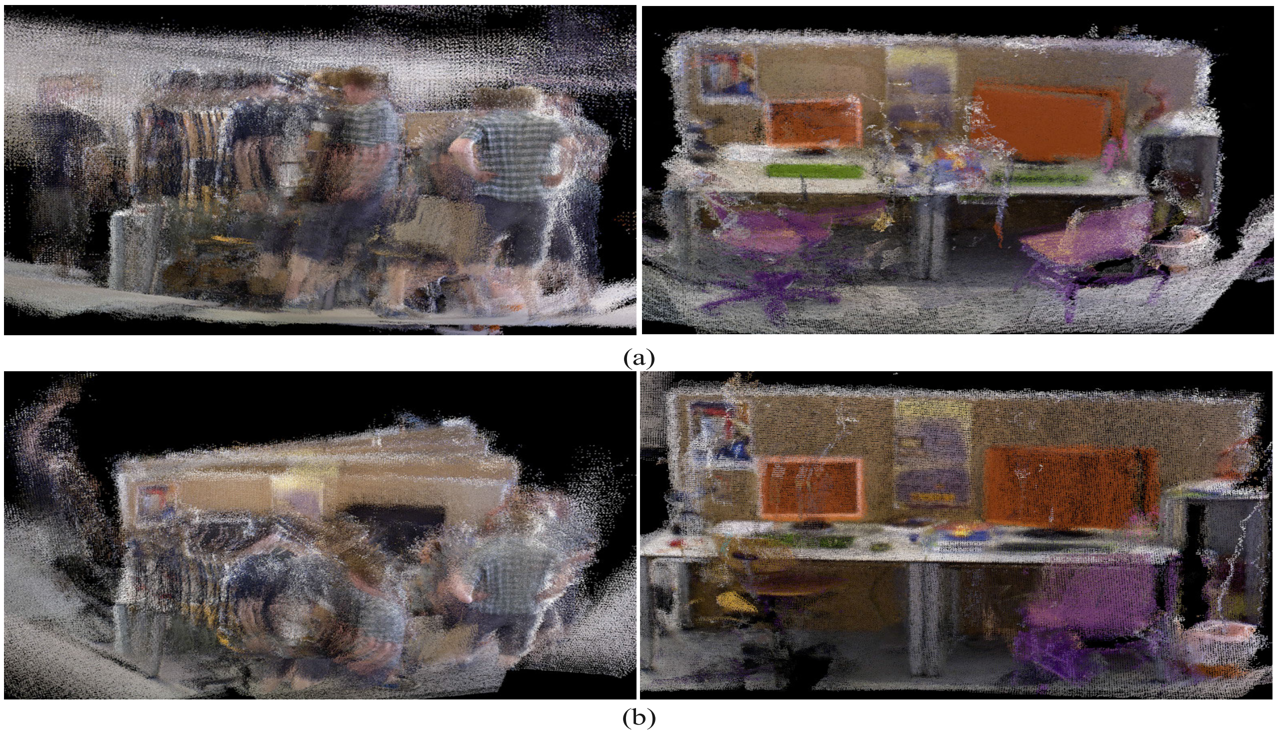


FIGURE 9. Dense 3D semantic map construction. (a) Sequences from fr3_w_h. (b) Sequences from fr3_w_x. Comparison of ORB-SLAM2 results (Left) and YG-SLAM results (Right) after dynamic object removal.

E. DENSE 3D SEMANTIC MAP CONSTRUCTION

In this section, we introduced the YG-SLAM method to effectively filter and eliminate dynamic objects from the environment, generating a dense point cloud without dynamic elements. We validated this approach on two sequences (fr3_w_h and fr3_w_x) from the TUM dataset. As illustrated

TABLE 5. Time evaluation.

Approach	Semantic Segmentation [ms]	Track Each Frame [ms]	Hardware Platform
DynaSLAM	195	>300	Nvidia Tesla M40 GPU
DS-SLAM	59.40	>65	Intel i7 CPU, P4000 GPU
YG-SLAM	21.10	42	Intel i7-11800H CPU, GTX 3060 GPU

in Fig.9, the left image demonstrates the result obtained using ORB-SLAM2 without dynamic object removal, exhibiting numerous double-shadow artifacts. In contrast, the right image displays the outcome after applying YG-SLAM for dynamic object removal, successfully eliminating all dynamic objects from the scene. This technique not only removes feature points on the dynamic mask in visual odometry, eliminating the impact of dynamic objects on pose estimation and improving trajectory accuracy but also employs a dynamic object mask during mapping, preventing the incorporation of moving targets into the map, thereby enhancing map accuracy.

V. CONCLUSION

This paper introduces a novel SLAM system called YG-SLAM, designed to mitigate the impact of dynamic objects on localization. YG-SLAM is built on the foundation of ORB-SLAM2 and comprises two designed threads: a tracking thread and an instance segmentation thread. For the instance segmentation thread, a lightweight version of YOLOv8n is proposed to provide essential semantic information in dynamic environments. By combining real-time semantic segmentation using YOLOv8 and improved motion consistency checks, the system filters out the dynamic elements in the scene, such as pedestrians, enabling SLAM to perceive environmental information. Subsequently, matching feature points are removed from the detected dynamic areas, enhancing robustness and accuracy in dynamic scenarios. The constructed point cloud and semantic map do not include dynamic point cloud information, resulting in minimal information redundancy and rich semantic data. Experiments were conducted on the TUM public dynamic dataset [52], and the results demonstrate significant improvements in localization accuracy compared to traditional methods in dynamic environments.

However, YG-SLAM still faces ongoing work. In the future, we plan to utilize the results of semantic segmentation to build a semantic octree map, improving the robot's ability to navigate around moving obstacles in dynamic environments to meet broader requirements. Additionally, the system's performance may be affected when objects in target detection appear irregularly, such as when the camera captures people walking at a 45-degree angle, and when there is a significant amount of occlusion. Future work will focus on addressing missed detection issues, which is one of our research directions.

ACKNOWLEDGMENT

The authors would like to convey their thanks to the editors and the unattributed reviewers for their insightful input and beneficial advice.

REFERENCES

[1] K. A. A. Mustafa, N. Botteghi, B. Sirmacek, M. Poel, and S. Stramigioli, "Towards continuous control for mobile robot navigation: A reinforcement learning and SLAM based approach," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. XLII-2/W13, pp. 857–863, 2019. [Online]. Available: https://isprs-archives.copernicus.org/articles/XLII-2-W13/857/2019/, doi: 10.5194/isprs-archives-XLII-2-W13-857-2019.
[2] F. Marques, P. Costa, F. Castro, and M. Parente, "Self-supervised subsea SLAM for autonomous operations," in *Proc. Offshore Technol. Conf.*, 2019, Art. no. D011S002R006.
[3] R. Liu, J. Zhang, K. Yin, J. Wu, R. Lin, and S. Chen, "Instant SLAM initialization for outdoor omnidirectional augmented reality," in *Proc. 31st Int. Conf. Comput. Animation Social Agents*. New York, NY, USA: Association for Computing Machinery, May 2018, p. 66, doi: 10.1145/3205326.3205359.
[4] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, "1-point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry," *J. Field Robot.*, vol. 27, no. 5, pp. 609–631, Sep. 2010.
[5] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Computer Vision—ECCV*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 834–849.
[6] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
[7] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.
[8] T. Qin, P. Li, and S. Shen, "VINS-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018, doi: 10.1109/TRO.2018.2853729.
[9] R. Wang, M. Schworer, and D. Cremers, "Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2017, pp. 3903–3911.
[10] G. Tian, L. Liu, J. Ri, Y. Liu, and Y. Sun, "ObjectFusion: An object detection and segmentation framework with RGB-D SLAM and convolutional neural networks," *Neurocomputing*, vol. 345, pp. 3–14, Jun. 2019.
[11] A. Singandhupe and H. M. La, "A review of SLAM techniques and security in autonomous driving," in *Proc. 3rd IEEE Int. Conf. Robotic Comput. (IRC)*, Feb. 2019, pp. 602–607.
[12] F. Zhong, S. Wang, Z. Zhang, C. Chen, and Y. Wang, "Detect-SLAM: Making object detection and SLAM mutually beneficial," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1001–1010.
[13] 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," *Robot. Auto. Syst.*, vol. 117, pp. 1–16, Jul. 2019.

[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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1168–1174.

[15] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, ''DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes,'' *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 4076–4083, Oct. 2018.

[16] N. Brasch, A. Bozic, J. Lallemand, and F. Tombari, ''Semantic monocular SLAM for highly dynamic environments,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 393–400.

[17] K. Wang, Y. Lin, L. Wang, L. Han, M. Hua, X. Wang, S. Lian, and B. Huang, ''A unified framework for mutual improvement of SLAM and semantic segmentation,'' in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 5224–5230.

[18] X. Yuan and S. Chen, ''SaD-SLAM: A visual SLAM based on semantic and depth information,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 4930–4935.

[19] J. Vincent, M. Labbé, J.-S. Lauzon, F. Grondin, P.-M. Comtois-Rivet, and F. Michaud, ''Dynamic object tracking and masking for visual SLAM,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 4974–4979.

[20] A. Li, J. Wang, M. Xu, and Z. Chen, ''DP-SLAM: A visual SLAM with moving probability towards dynamic environments,'' *Inf. Sci.*, vol. 556, pp. 128–142, May 2021.

[21] T. Ji, C. Wang, and L. Xie, ''Towards real-time semantic RGB-D SLAM in dynamic environments,'' in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 11175–11181.

[22] Y. Fan, Q. Zhang, Y. Tang, S. Liu, and H. Han, ''Blitz-SLAM: A semantic SLAM in dynamic environments,'' *Pattern Recognit.*, vol. 121, Jan. 2022, Art. no. 108225. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320321004064

[23] J. Cheng, Z. Wang, H. Zhou, L. Li, and J. Yao, ''DM-SLAM: A feature-based SLAM system for rigid dynamic scenes,'' *ISPRS Int. J. Geo-Inf.*, vol. 9, no. 4, p. 202, Mar. 2020.

[24] H. Liu, G. Liu, G. Tian, S. Xin, and Z. Ji, ''Visual SLAM based on dynamic object removal,'' in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2019, pp. 596–601.

[25] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger, ''MID-fusion: Octree-based object-level multi-instance dynamic SLAM,'' in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 5231–5237.

[26] S. Wen, P. Li, Y. Zhao, and Z. Wang, ''Semantic visual SLAM in dynamic environment,'' *Auto. Robots*, vol. 45, pp. 493–504, May 2021.

[27] K. M. Judd, J. D. Gammell, and P. Newman, ''Multimotion visual odometry (MVO): Simultaneous estimation of camera and third-party motions,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 3949–3956.

[28] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[29] S. Baker and I. Matthews, ''Lucas-Kanade 20 years on: A unifying framework,'' *Int. J. Comput. Vis.*, vol. 56, no. 3, pp. 221–255, Feb. 2004.

[30] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, ''Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,'' *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[31] S. Li and D. Lee, ''RGB-D SLAM in dynamic environments using static point weighting,'' *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 2263–2270, Oct. 2017.

[32] R. Wang, W. Wan, Y. Wang, and K. Di, ''A new RGB-D SLAM method with moving object detection for dynamic indoor scenes,'' *Remote Sens.*, vol. 11, no. 10, p. 1143, May 2019. [Online]. Available: https://www.mdpi.com/2072-4292/11/10/1143

[33] J. Cheng, C. Wang, and M. Q.-H. Meng, ''Robust visual localization in dynamic environments based on sparse motion removal,'' *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 2, pp. 658–669, Apr. 2020.

[34] W. Dai, Y. Zhang, P. Li, Z. Fang, and S. Scherer, ''RGB-D SLAM in dynamic environments using point correlations,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 373–389, Jan. 2022.

[35] E. Palazzolo, J. Behley, P. Lottes, P. Giguère, and C. Stachniss, ''ReFusion: 3D reconstruction in dynamic environments for RGB-D cameras exploiting residuals,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 7855–7862.

[36] B. K. P. Horn and B. G. Schunck, ''Determining optical flow,'' *Artif. Intell.*, vol. 17, nos. 1–3, pp. 185–203, Aug. 1981.

[37] M. Derome, A. Plyer, M. Sanfourche, and G. Le Besnerais, ''Real-time mobile object detection using stereo,'' in *Proc. 13th Int. Conf. Control Autom. Robot. Vis. (ICARCV)*, Dec. 2014, pp. 1021–1026.

[38] M. Derome, A. Plyer, M. Sanfourche, and G. L. Besnerais, ''Moving object detection in real-time using stereo from a mobile platform,'' *Unmanned Syst.*, vol. 3, no. 4, pp. 253–266, Oct. 2015, doi: 10.1142/s2301385015400026.

[39] M. C. Bakkay, M. Arafa, and E. Zagrouba, ''Dense 3D SLAM in dynamic scenes using Kinect,'' in *Pattern Recognition and Image Analysis*. Santiago de Compostela, Spain: Springer, 2015, pp. 121–129.

[40] T. Zhang, H. Zhang, Y. Li, Y. Nakamura, and L. Zhang, ''FlowFusion: Dynamic dense RGB-D SLAM based on optical flow,'' in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 7322–7328.

[41] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, ''PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8934–8943.

[42] V. Badrinarayanan, A. Kendall, and R. Cipolla, ''SegNet: A deep convolutional encoder–decoder architecture for image segmentation,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[43] M. Runz, M. Buffier, and L. Agapito, ''MaskFusion: Real-time recognition, tracking and reconstruction of multiple moving objects,'' in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Oct. 2018, pp. 10–20.

[44] K. He, G. Gkioxari, P. Dollár, and R. Girshick, ''Mask R-CNN,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020.

[45] N. Dvornik, K. Shmelkov, J. Mairal, and C. Schmid, ''BlitzNet: A real-time deep network for scene understanding,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4174–4182.

[46] L. Yan, X. Hu, L. Zhao, Y. Chen, P. Wei, and H. Xie, ''DGS-SLAM: A fast and robust RGBD SLAM in dynamic environments combined by geometric and semantic information,'' *Remote Sens.*, vol. 14, no. 3, p. 795, Feb. 2022. [Online]. Available: https://www.mdpi.com/2072-4292/14/3/795

[47] R. Long, C. Rauch, V. Ivan, T. Lun Lam, and S. Vijayakumar, ''RGB-D-inertial SLAM in indoor dynamic environments with long-term large occlusion,'' 2023, *arXiv:2303.13316*.

[48] L. Tian, Y. Yan, and H. Li, ''SVD-SLAM: Stereo visual SLAM algorithm based on dynamic feature filtering for autonomous driving,'' *Electronics*, vol. 12, no. 8, p. 1883, Apr. 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/8/1883

[49] R. Liang, J. Yuan, B. Kuang, Q. Liu, and Z. Guo, ''DIG-SLAM: An accurate RGB-D SLAM based on instance segmentation and geometric clustering for dynamic indoor scenes,'' *Meas. Sci. Technol.*, vol. 35, no. 1, Sep. 2023, Art. no. 015401, doi: 10.1088/1361-6501/acfb2d.

[50] Y. Fang, Z. Xie, K. Chen, G. Huang, R. Zarei, and Y. Xie, ''DYS-SLAM: A real-time RGBD SLAM combined with optical flow and semantic information in a dynamic environment,'' *J. Intell. Fuzzy Syst.*, vol. 14, pp. 1–19, Dec. 2023.

[51] M. A. Fischler and R. Bolles, ''Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,'' *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[52] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, ''A benchmark for the evaluation of RGB-D SLAM systems,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 573–580.

GUOMING CHU received the B.E. degree in automation from Xihua University, Chengdu, China, in 2017. He is currently pursuing the M.S. degree with the Sichuan University of Science and Engineering, Yibin, China. Subsequently, he has gained work experience with China Railway Tunnel Group Company Ltd., from 2017 to 2021. His research interests include image processing, deep learning, and visual simultaneous localization and mapping (SLAM). In July 2023, he received the ''Second Prize for Teams'' at the 18th China Graduate Electronic Design Competition, Southwest Regional Competition.

YAN PENG received the B.S. degree in applied mathematics from Sichuan University, Chengdu, China, in 1991, and the M.S. and Ph.D. degrees in metallurgical machinery and mechanical manufacturing and automation from Chongqi, China, in 1997 and 2001, respectively. He is currently the Director of the Engineering Practice Center, Sichuan University of Science and Engineering, and the Key Laboratory of Enterprise Informatization and Internet of Things Measurement and Control in Sichuan Province Universities. His research interests include the Internet of Things technology and applications, machine learning, natural language processing, expert systems, and intelligent information processing.

XUHONG LUO received the B.E. degree in computer science and technology from Jiangsu Ocean University, Jiangsu, China, in 2021. He is currently pursuing the M.S. degree with the Sichuan University of Science and Engineering, Yibin, China. His research interests include image processing, deep learning, and YOLO algorithm.

JING GONG received the bachelor's degree in communication engineering from Chengdu Technological University, in 2021. He is currently pursuing the master's degree with the Sichuan University of Science and Engineering, Yibin, China. His research interests include neural networks and network resource allocation.