Semantic SLAM for mobile robots in dynamic environments based on visual camera sensors

OPEN ACCESS
IOP Publishing
Measurement Science and Technology
Meas. Sci. Technol. **34** (2023) 085202 (12pp)
https://doi.org/10.1088/1361-6501/acd1a4

Semantic SLAM for mobile robots in dynamic environments based on visual camera sensors

Qi Zhang[1,*] **and Changdi Li**[2]

[1] School of Computing Science, University of Glasgow, Glasgow G12 8QQ, United Kingdom
[2] College of Control Science and Engineering, Zhe Jiang University, Hang Zhou 310000, People's Republic of China

E-mail: zhangqi_research@163.com

Received 24 January 2023, revised 14 April 2023
Accepted for publication 2 May 2023
Published 11 May 2023

Abstract

Visual simultaneous localization and mapping (vSLAM) is inherently constrained by the static world assumption, which renders success in the presence of dynamic objects rather challenging. In this paper, we propose a real-time semantic vSLAM system designed for both indoor and outdoor dynamic environments. By employing object detection, we identify 80 categories and utilize motion consistency checks to pinpoint outliers in each image. Distinct methods are presented for examining the motion states of humans and other objects. For detected humans, an algorithm is introduced to assess whether an individual is seated, subsequently dividing the bounding boxes of seated individuals into two parts based on human body proportions. We then use the same threshold values for standing individuals to determine the states of the two boxes belonging to seated individuals. For non-human objects, we propose an algorithm capable of automatically adjusting the threshold values for different bounding boxes, thereby ensuring consistent detection performance across various objects. Ultimately, we retain points within static boxes contained in dynamic boxes while eliminating other points in dynamic boxes to benefit from a larger number of detected categories. Our SLAM is evaluated on indoor TUM and Bonn RGB-D datasets, with further testing conducted on the outdoor stereo KITTI dataset. The results reveal that our SLAM outperforms most SLAM systems in dynamic environments. Moreover, we test our system in real-world environments with a monocular camera, demonstrating its robustness and universality across diverse settings.

Keywords: localization, SLAM, deep learning

(Some figures may appear in colour only in the online journal)

1. Introduction

Simultaneous localization and mapping (SLAM) plays an integral role in robot vision. It can approximate the poses of the camera and rebuild the unknown environment through various sensors. As camera-based systems are cheaper than other sensor-based systems [1–3], many visual SLAM (vSLAM) systems with good performance have been proposed [4–6]. In particular, the method that works for all categories of cameras is the cheapest. Whereas, if the method is limited by depth information, only relatively expensive RGB-D cameras can be used. However, most SLAM systems are constrained by the static environment assumption and disturbed by dynamic objects in the real world, causing many bad or unstable data associations.

* Author to whom any correspondence should be addressed.

For such problems, scholars usually use the idea of tracking only stable static features. Some use geometric-based approaches, such as RANSAC [7] (random sample consensus), to remove mismatches in the static or dynamic scene. Although this method can remove outliers, it is useless when dynamic objects occupy most of the views. Recently, more researchers have focused on combining deep learning with the traditional geometric methods to deal with dynamic objects. Their main idea is to use semantic segmentation or object detection to get masks or bounding boxes of objects defined as movable and remove the feature points of those objects through geometric information. The latest research result, Crowd-SLAM [8], shows that the method based on object detection can better process no predefined moving objects than those based on semantic segmentation. However, there are too many feature points in the bounding boxes. If all of them are erased, there will not be enough data associations for pose estimation, and the SLAM system will fail. At the same time, semantic segmentation algorithms such as SegNet [9] or Mask-RCNN [10] are difficult to achieve a good balance in segmentation accuracy, system load, and the number of classes detected. In other words, the system can not run in real-time when it is accurate enough and can detect many classes of objects.

This paper proposes a real-time semantic vSLAM system to solve the above problems. Our SLAM adopted ORB-SLAM3, a state-of-the-art SLAM system that added a multiple map system to increase the performance in the large-scale scene. We deployed the dynamic point culling algorithm in the front end of the SLAM system. First, we extracted the semantic information of 80 different object classes in the environment using the TensorRT optimized YOLOX [11], known as the latest one-stage detector and can recognize many kinds of objects with high accuracy in real-time. For the detected person, we propose an algorithm for judging whether the person detected is sitting, called Nine Head Body Checking (NHBC). Then we divide the bounding box belonging to a sitting person into upper body and lower body parts in a certain proportion. Meanwhile, we use the moving consistency checking to determine the motion state of points in all the boxes. If an outlier is contained in the human box, the box will be determined as dynamic. In order to make our SLAM perform better on different occasions, we propose an algorithm to check other classes except for humans, adaptive threshold adjusting (ATA). The box will be judged as moving if the number of outliers inside it exceeds the threshold, which is adjusted automatically for different objects. Finally, to make the detected classes benefit the system's accuracy, we will preserve feature points of static boxes within or outside the dynamic box and eliminate other points in the dynamic box. This new dynamic point culling algorithm has increased the number of stable static points. As shown in figure 1, it can be seen that the lower body points and the static objects in the frame have not been culled, even though some objects are partially obscured. The contribution of this letter has three points:

- We propose a real-time dynamic semantic vSLAM algorithm with high accuracy in both outdoor and indoor environments, which can be deployed on various cameras and is not limited by depth information.
- A dynamic point culled algorithm benefits from more detected classes, robust to known and limited unknown dynamic objects in different environments.
- An algorithm almost unaffected by noise points for judging moving bounding boxes is raised, which considers that all detected objects are possible to move and checks the motion states of people with different postures and objects using different methods.

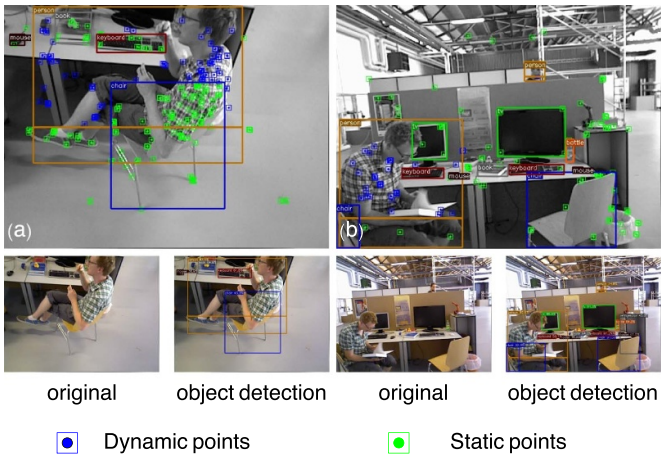


Figure 1. A general description of detecting moving points in our system. In example (a), the chair, book, and keyboard points are marked as static, and the human's upper body points are judged as dynamic by combining the semantic and geometrical information in our method. In case (b), the points associated with a moving book without semantic prior are also determined as dynamic.

The rest of the paper is built as follows. The next section (i.e. section 2) describes works on improving SLAM in dynamic environments. Section 3 displays a detailed approach for improving dynamic SLAM using object detection. Finally, sections 4 and 5 show the results obtained from experiments and the conclusions drawn from the results, respectively.

2. Related work

2.1. Dynamic SLAM improved by geometry method

Sun *et al* [12] use particle filters to track and filter motion patches in images. However, the discontinuous movement will lead to tracking failure. Zhang *et al* [13] used depth and intensity information to approximate the camera's ego-motion, which is then utilized to detect moving features and is non-sensitive to slight motions. Dai *et al* [14] propose to utilize point correlations to identify static and dynamic map points. Scona *et al* [15] simultaneously approximate the camera motion and probabilistic segmentation of the current RGB-D frame. Du *et al* [16] use long-term consistency via conditional random fields to detect dynamic components. The performance is not very good for objects that remain static for a long time before moving. The above methods only work on RGB-D cameras, and most of them are constantly disturbed

by slow objects and blocked objects. By contrast, our method can handle the two kinds of objects.

2.2. Dynamic SLAM improved by the semantic method

Semantic segmentation or object detection can give prior information about moving objects to the SLAM system. But these works all assume that fixed few classes have the possibility to move in the scene. Bescos *et al* [17] raised DynaSLAM. They use Mask-R-CNN to remove the points of objects marked as potentially dynamic in 20 classes without checking their motion state and a multi-view geometry method to manage unknown dynamic objects. However, it is not a real-time system. Redmon and Farhadi raised the CrowdSLAM, which uses YOLOv3 [18] to detect and directly remove the key points only in one class, people's bounding box. Ji *et al* [19] used semantic segmentation to remove objects defined as highly movable in 20 classes and then detect unknown dynamic objects by the reprojection errors of clustering the depth image. In the above method, whether the box is moving or not, its key points will be culled, significantly reducing the data associations and improving the probability of tracking failure.

Furthermore, some researchers determine whether the segmented objects are moving or not. Yu *et al* [20] raised DS-SLAM. They combined SegNet and optical flow to remove the moving points of humans in the image. Cheng *et al* [21] raised DM-SLAM, which utilizes the Mask-R-CNN and epipolar geometry to detect the dynamic points of the few potential moving objects in the scene. Fan *et al* [22] utilize the BlitzNet [23] to segment objects and the epipolar constraint to remove outliers in the dynamic mask areas of a few classes marked as movable. These works are not robust to unknown objects, and the fixed threshold will make the algorithm sensitive to some objects and insensitive to others.

Many studies recently focused on the ORB-SLAM3, a stable SLAM system proposed recently. Based on the ORB-SLAM3, Hu *et al* [24] use the DeepLab v3+ [25] to dynamic segment objects and filter them with multi-view geometry, which is not capable of handling unknown objects. Liu and Miura [26] use moving probability to update and propagate semantic information to filter out moving points in tracking. They only segment 20 classes of objects and do not apply their method to other types of cameras. Although the above methods rely on semantic information, they deal with people in different postures and other objects in the same way.

3. System overview

We use semantic and geometric methods to filter out dynamic feature points in RGB images. Firstly, we use the learning-based method to get the object's bounding box. At the same time, the outliers in the image are obtained by the technique of motion consistency check. Then, the efficient algorithm we proposed detects and removes the moving points by combining the two kinds of information. Our system is based on ORB-SLAM3, a feature points-based system for static environments. In particular, the semantic module of the system refers to and improves the work of CrowdSLAM, while the geometric module refers to the work of DS-SLAM. The framework of the system proposed in our paper can be seen in figure 2.

3.1. Semantic module

The semantic module predicts the bounding boxes of different objects in RGB images using deep learning-based methods. We adopted the YOLOX, which performs better than the latest version of the YOLO Series. It means that the YOLOX has excellent accuracy while maintaining a high computational speed. Furthermore, to reduce the processing time, we use the GPU acceleration method, TensorRT [27], to optimize the network of the YOLOX model. Different from other SLAM methods, we believe that in complex scenarios, all objects have the possibility of moving. At the same time, this also benefits the construction of semantic maps later. So, to identify as many categories as possible, We pre-train the YOLOX model on the COCO dataset [28], which contains 80 classes of objects. Then, Each box will be checked for motion consistency combined with an adaptive threshold algorithm to ascertain whether the object is dynamic.

However, the CrowdSLAM referred to in our system uses YOLOv3, which only regards people as the potential moving object, making its application scenarios more limited. Moreover, it does not use the geometric method to filter the bounding box, eliminating too many key points.

3.2. Geometry module

The module is designed for obtaining the outliers in the input image. We adopt the motion consistency checking method proposed by DS-SLAM. Like DS-SLAM, we match Harris corners by calculating the optical flow pyramid. If the distance between the matching point and the pixel edge is very small or the pixel blocks in the center of the matching pair are very different, it will be abandoned. Then we calculate the distance between a remaining point successfully matched and the epipolar line corresponding to it. The point will be judged outlier if the distance exceeds our defined threshold.

3.3. Judging moving boxes

We use RANSAC with the most inliers to find the fundamental matrix to calculate the polar line of the current frame. Specifically, The fundamental matrix maps the points in the last frame to the search domain corresponding to them in the current frame, namely, the epipolar line. Make p_1, p_2 represent the points matched successfully in the last frame and current frame, respectively, and P_1, P_2 denote their homogeneous coordinate form:

$$\begin{aligned} P_1 = [u_1, v_1, 1], P_2 = [u_2, v_2, 1], \\ p_1 = [u_1, v_1], p_2 = [u_2, v_2] \end{aligned} \tag{1}$$

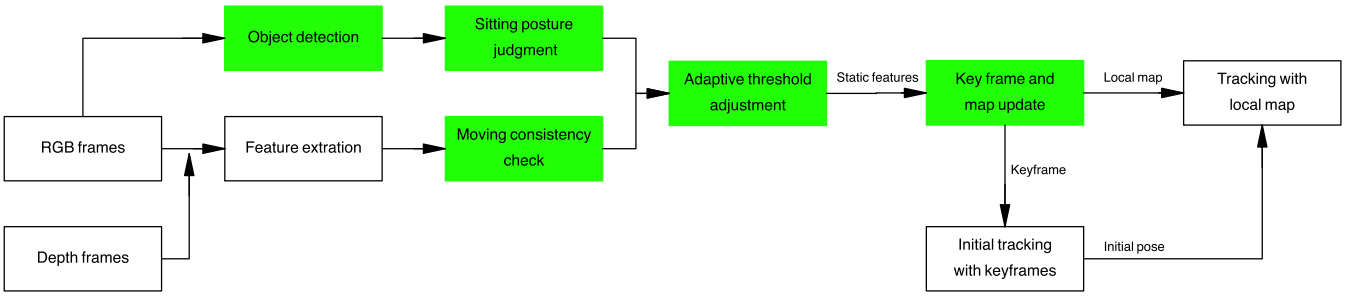


Figure 2. The general framework of our SLAM system. The black areas are the original ORB-SLAM3 tracking module, and the green areas are our added or modified modules, consisting of a module using the semantic method, a module using the geometry method, a moving box judging module, and a culling dynamic points module. The semantic module detects the objects in the new RGB image just arriving and predicts the bounding boxes of them. The geometry module checks the moving consistency of these boxes. Then, the moving boxes judging module can determine the motion state of boxes by combining the above information. Finally, the culling dynamic points module removes the moving points according to the state of bounding boxes, and the rest points are used for pose estimation.

Algorithm 1. Moving boxes judgment algorithm.

Input: Bounding boxes set BH_n of person, bounding boxes set BO_n of other objects;
Output: Dynamic bounding boxes DB_n, static bounding boxes SB_n;

function JUDGING(Set)
  $o \leftarrow$ total number of outliers in Set;
  if $o > 0$ **then**
    $Set \rightarrow DB_n$;
  else
    $Set \rightarrow SB_n$;
  end if
end function
for each bounding box $BH_n(i)$ of person **do**
  if $\frac{W_{BH_n(i)}}{H_{BH_n(i)}} < 3$ **then**
    Divide the box into $\frac{0.3}{1.3}$ and $\frac{1}{1.3}$ upper and lower parts BH_U and BH_L;
    JUGING(BH_U);
    JUGING(BH_L);
  else
    JUGING($BH_n(i)$);
  end if
end for
for each bounding box $BO_n(i)$ of other objects **do**
  $s \leftarrow$ total number of key points tracked by the optical flow in $BO_n(i)$;
  $o \leftarrow$ total number of outliers in $BO_n(i)$;
  if $\frac{o}{s} > \epsilon$ **then**
    $BO_n(i) \rightarrow DB_n$;
  else
    $BO_n(i) \rightarrow SB_n$;
  end if
end for

where u, v are the coordinate values in the image. The L_1 represents the epipolar line, and its calculation method is as follows:

$$L_1 = \begin{vmatrix} X \\ Y \\ Z \end{vmatrix} = FP_1 = F \begin{vmatrix} u_1 \\ v_1 \\ 1 \end{vmatrix} \tag{2}$$

where X, Y, Z denote line vector, and F denotes fundamental matrix. The method we calculate the distance between the matched point and its corresponding epipolar line is as follows:

$$D = \frac{|P_2^T F P_1|}{\sqrt{\|X^2\| + \|Y^2\|}} \tag{3}$$

where D denotes the distance. If the value of D exceeds the preset threshold, the feature point will be judged as an outlier.

The module utilizes semantic and geometry information to determine whether a bounding box is moving. Firstly, we use a unique method to handle the bounding box belonging to people. In daily life, sitting people only have half of their body in motion, while standing people's whole body is moving. Therefore, we design a special algorithm, NHBC, to judge whether a person is sitting. In the field of painting, painters will use the length and width of heads to draw the body of the human. For example, when people sit, the width is two people's heads, and the length is five. The basic theory is shown in figure 3. The length to width ratio of people in standing posture is $\frac{9}{2}$, $\frac{8}{2}$, and $\frac{7}{2}$ and those of sitting is $\frac{5}{2}$. Hence, the length-to-width ratio of sitting people is less than 3; When they stand, the length-width ratio is usually 4, greater than 3. Therefore, we mark the bounding box belonging to people whose ratio is less than three as sitting posture. The length-to-width ratio is determined as follows:

$$R_i = \frac{W_i}{H_i} \tag{4}$$

where R_i represents the length-to-width ratio of the bounding box i. The width and height of i are denoted as W_i, H_i, respectively. In order to divide the human hand into the upper body at any time, according to the human activity in the dataset, when the length to width ratio of the human's box is less than 3, we divide the box into $\frac{1}{1.3}$ and $\frac{0.3}{1.3}$ two parts. As shown in figures 4(a)–(h), the hands of people are always contained in the upper part. It is noteworthy that if only local parts of the human body are observed, NHBC will fail. At this time, taking figures 4(e) and (f) as examples, people occupy most of the views, and their movement will be easy to be checked. It alleviates the shortage of the algorithm to some extent.

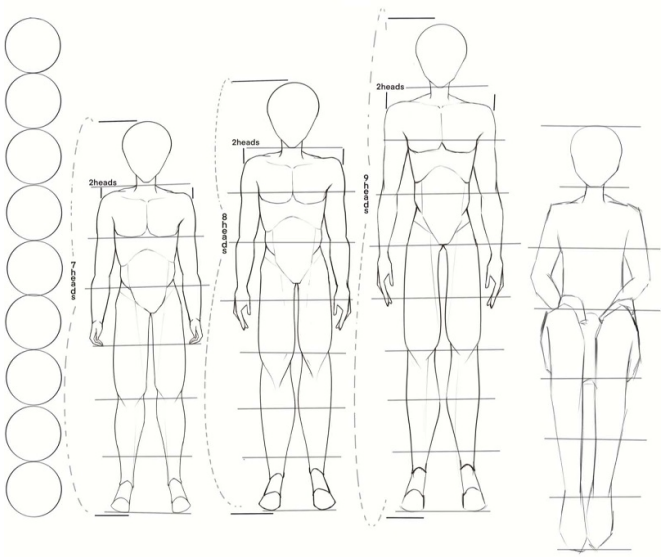


Figure 3. Schematic diagram of different human body proportions in standing and sitting in painting field. From left to right, there are standing postures of 7 heads, 8 heads, 9 heads high, and a sitting posture of 7 heads high, respectively. At the same time, these human postures are all two head widths.

Then, the threshold of people and objects is determined separately. It is because people and objects move in different ways. Precisely, people often move local joints. Namely, only one or two points are dynamic. As shown in figures 4(a) and (b), the person in the figure has only hand joints moving, and only one or two feature point is dynamic. Therefore, when an outlier is checked out in the person's box, the box will be marked as dynamic. Other objects do not always move a tiny part like a human. So, more feature points should be considered to judge. On the other hand, some objects are extracted with few feature points, and others are extracted with many points. In that case, if a fixed threshold is set, it will not be suitable for all objects.

Therefore, we propose the ATA. Firstly, We divide the outliers in the box by the total number of feature points tracked by the optical flow in the same box. Then, if the calculation result is larger than the defined value, the bounding box will be marked as moving. Specifically, The algorithm is not only robust for different kinds of objects but also solves the problem that objects are mistakenly considered to be dynamic when the object is occluded or the camera moves itself to some extent. Finally, because false-negative results (namely dynamic objects marked as static) have a terrible impact on the SLAM system, we set a relatively low threshold. For the upper and lower two parts, or the whole body box of moving people, we set the threshold to one outlier, and for judging dynamic objects, the threshold is defined as 40% outliers in the box. The percentage setting allows the threshold to be automatically adjusted according to different object boxes. Meanwhile, Objects less than 40% occluded will not be judged as dynamic. Moreover, Judging based on multiple outliers can effectively eliminate the influence of noise and camera self-movement. The moving boxes judgment algorithm is displayed in algorithm 1, and the ε represents a preset threshold value. The width of the bounding box $BH_n(i)$ is defined as $W_{BH_n(i)}$ and the height of the box is denoted as $H_{BH_n(i)}$.

Algorithm 2. Moving boxes judgment algorithm.

Input: Bounding boxes set BH_n of person, bounding boxes set BO_n of other objects;
Output: Dynamic bounding boxes DB_n, static bounding boxes SB_n;

function JUDGING(Set)
  $o \leftarrow$ total number of outliers in Set;
  if $o > 0$ **then**
    $Set \rightarrow DB_n$;
  else
    $Set \rightarrow SB_n$;
  end if
end function
for each bounding box $BH_n(i)$ of person **do**
  if $\frac{W_{BH_n(i)}}{H_{BH_n(i)}} < 3$ **then**
    Divide the box into $\frac{0.3}{1.3}$ and $\frac{1}{1.3}$ upper and lower parts BH_U and BH_L;
    JUGING(BH_U);
    JUGING(BH_L);
  else
    JUGING($BH_n(i)$);
  end if
end for
for each bounding box $BO_n(i)$ of other objects **do**
  $s \leftarrow$ total number of key points tracked by the optical flow in $BO_n(i)$;
  $o \leftarrow$ total number of outliers in $BO_n(i)$;
  if $\frac{o}{s} > \epsilon$ **then**
    $BO_n(i) \rightarrow DB_n$;
  else
    $BO_n(i) \rightarrow SB_n$;
  end if
end for

3.4. Culling dynamic points

After each box is marked, due to the ambiguity of the bounding box, we only remove the points, which are outside the static box and in the dynamic box at the same time. As shown in figures 4(a)–(c), (f) and (g), the key points in the static box belonging to a person's box or intersecting with the human box are preserved. This means that the more static boxes make the more stable static points, namely, more accuracy for the system. The dynamic points culled algorithm is shown in algorithm 2.

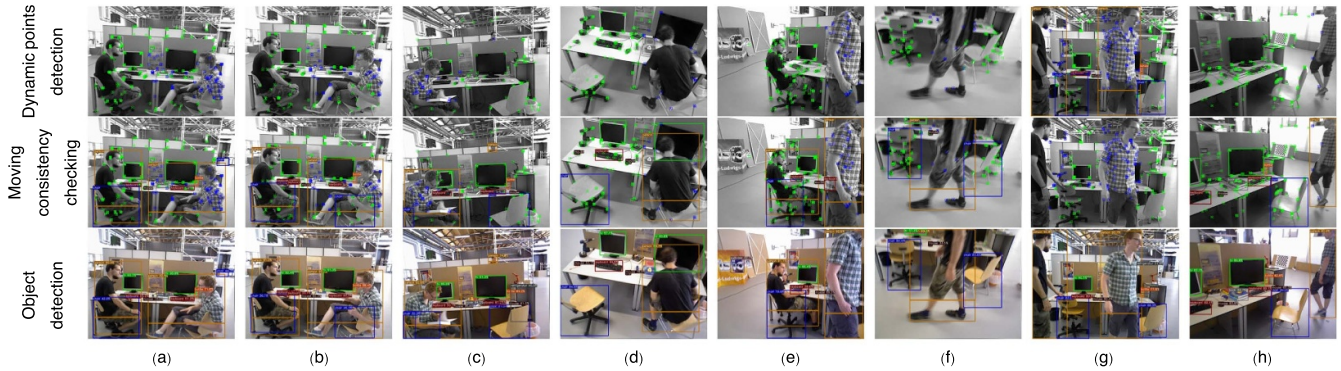


Figure 4. Examples of moving object detection. The first line presents bounding boxes detected by the proposed semantic module. The second-row results from judging moving boxes and detecting dynamic points, with bounding boxes shown. The third row shows the same results as the second but without the boxes shown. The dynamic points are marked as blue and the static points as green.

Algorithm 3. Dynamic points culled algorithm.

Input: Dynamic bounding boxes DB_n, static bounding boxes SB_n, The set of key points for the current frame K_n;
Output: The set of key points judged as static for the current frame S_n;

$bool_a \leftarrow 0$;
for each key point (u_n, v_n) in K_n of person **do**
  for each bounding box $DB_n(i)$ of person **do**
    if (u_n, v_n) in $DB_n(i)$ **then**
      $bool_a \leftarrow 1$;
    end if
  end for
  for each bounding box $BO_n(i)$ of other objects **do**
    if (u_n, v_n) in $BO_n(i)$ **then**
      $bool_a \leftarrow 0$;
    end if
  end for
  if $bool_a = 0$ **then**
    $S_n \leftarrow (u_n, v_n)$;
  end if
end for

4. Experiments and results

4.1. Overview

We assess our system utilizing RGB-D and stereo camera datasets. To begin with, we take advantage of the TUM dataset's [29] generalization, employing three distinct groups of works for a comprehensive comparison. The first group contains systems based on ORB-SLAM3, which shows the effectiveness of our method on an identical baseline. The second group consists of two works similar to our approach; one employs the epipolar constraint as our method, while the other ingeniously amalgamates vSLAM with the YOLOv3. The third group represents cutting-edge research that has recently yielded promising results through geometric and semantic methodologies.

Subsequently, we employ the Bonn dataset [30] to examine our system's performance in increasingly intricate scenarios, comparing it with the three most advanced systems currently available. Moreover, we utilize the KITTI dataset [31] to measure our stereo camera mode system's efficacy in outdoor environments. To further enhance our comparative analysis, we incorporate the latest visual odometry (VO) and visual inertial odometry (VIO) systems, comparing them with our work.

Finally, the runtime analysis demonstrates the excellent real-time performance of our system. Simultaneously, we deploy our monocular camera mode system in a real-world environment to showcase its remarkable robustness.

4.2. Implementation

We utilize a laptop equipped with AMD Ryzen 7 5800H CPU, NVIDIA GeForce RTX 3070 GPU, and 16G RAM to conduct experiments on Ubuntu 20.04. At the same time, we use the YOLOX-s model for experiments, reaching a good balance between accuracy and real-time performance. Then, we test our system in a real-world environment with the monocular camera of our laptop, Lenovo R9000P.

4.3. TUM dataset

The TUM RGB-D dataset contains many image sequences caught through an RGB-D camera in a dynamic environment, as well as accurate ground truth trajectories and camera parameters. Dynamic SLAM is mainly for improving the SLAM system in high dynamic scenes, so we selected four sets of high dynamic sequences about walking: two people walk around or change the chair's location. At the same time, to test the system's performance under a low dynamic environment to a certain extent, We select two sets of low dynamic sitting sequences, that is, two people sit at the table to chat or do some small actions. The w/half, w/rpy, w/static, w/xyz, s/half, and s/xyz denote six groups of image sequences to express them easily, where w and s indicate walking sequences and siting sequences and half, rpy, static, and xyz represent different motion modes of the camera respectively.

4.4. Bonn dataset

The Bonn RGB-D Dynamic Dataset, provided by the University of Bonn in 2019, is a collection of 24 dynamic sequences

designed for evaluating RGB-D SLAM systems. The dataset acquires RGB-D images using the ASUS Xtion Pro LIVE camera and obtains the ground truth with the Optitrack Prime 13 motion capture system. To ensure a comprehensive evaluation of our system, we conduct experiments on 15 sequences, excluding those where no static points in the entire view and those with repetitive content.

Among these selected sequences, the 'crowd' sequence illustrates a scene where five individuals walk randomly in a room. The 'moving_nonobstructing_box2' sequence showcases a person moving a box from the floor to a table. The 'placing_obstructing _box' sequence portrays two people carrying a large box, approximately half a person's height and width, and placing it on the ground. The 'placing_nonobstructing_box' sequence displays a person transporting a small box and setting it on the floor. The 'person_tracking' sequence captures a scenario where the camera tracks a walking person. The 'balloon' sequence presents a scene where a person continuously hits a balloon, preventing it from falling. The 'balloon_tracking' sequence exhibits a person lifting a balloon and letting it fall to the ground. These complex scenarios further evaluate our SLAM system's performance in various conditions.

4.5. KITTI dataset

The KITTI dataset provides accurate ground truth trajectories, camera parameters, and object annotations for various tasks related to autonomous driving. Within this dataset, 22 stereo image sequences are specifically designed to evaluate the performance of vSLAM algorithms. These subsets are recorded by a vehicle in motion, covering a total distance of 39.2 km, with each sequence depicting distinct driving situations.

Our paper focuses on sequences 00–10, which offer ground truth information. These selected sequences encompass various road environments, including urban, suburban, and highway settings, varying traffic densities, and many lighting conditions. By analyzing these sequences, we can comprehensively evaluate the performance of our vSLAM system across a wide array of real-world driving scenarios.

4.6. Evaluation metrics

In the experiment, Our paper utilizes the absolute trajectory error (ATE) and the relative pose error (RPE) to measure the experimental results. Let $E_1,\ldots,E_n \in SE(3)$ represent the sequence of the approximately calculated poses, and $G_1,\ldots,G_n \in SE(3)$ represent the ground truth. The ATE A_t at time step t can be calculated by:

$$A_t = E_t^{-1} S G_t \tag{5}$$

where S represents the rigid-body transformation aline the approximately calculated trajectory with the ground truth with the same scale. The ATE can reflect the overall consistency of the approximately calculated trajectory, and the RPE is designed to estimate that local accuracy at a fixed time Δ. The method of calculating RPE R_t at time step t is as follows:

$$R_t = (E_t^{-1} E_{t+\Delta})^{-1} (G_t^{-1} G_{t+\Delta}). \tag{6}$$

4.7. Comparison with state-of-the-arts

4.7.1. TUM dataset. We have a comparison between our system and three groups of advanced dynamic SLAM systems. In each group, we give each system's RMSE and S.D of ATE to measure its global robustness and stability and use the RMSE and S.D of translational RPE to measure the local performance. We use t.RPE to represent translational RPE.

The first group includes two dynamic SLAM systems based on ORB-SLAM3. The ORB-SLAM3 is shown by O3. Meanwhile, the RDS represents RDS-SLAM [26], and the DeepLab represents the method proposed by Hu *et al* [24]. As presented in table 1, in ATE, the method proposed by us has nearly achieved the best performance in all sequences except the w/rpy sequence. Although DeepLab's performance in w/rpy is better than ours, it is t.RPE is not good enough in every sequence. Meanwhile, our method has significant advantages in APE, compared with ORB-SLAM3. As shown in figure 5, the estimated trajectories by our system are more accurate than ORB-SLAM3.

The second group mainly includes the SLAM systems referred to in this paper, as well as a not up-to-date but very authoritative method. In this group, the DS represents DS-SLAM [20], the Crowd represents Crowd-SLAM [8], and the Dynamic represents Dyna-SLAM [17]. Our semantic and geometric modules are based on the first two systems, and the performance of the last system is compared in many articles. As presented in table 2, our SLAM achieves better results than other systems, reflecting the robustness of our improved method. But our system's ATE of w/rpy sequence is also slightly less than DS-SLAM and Crowd-SLAM.

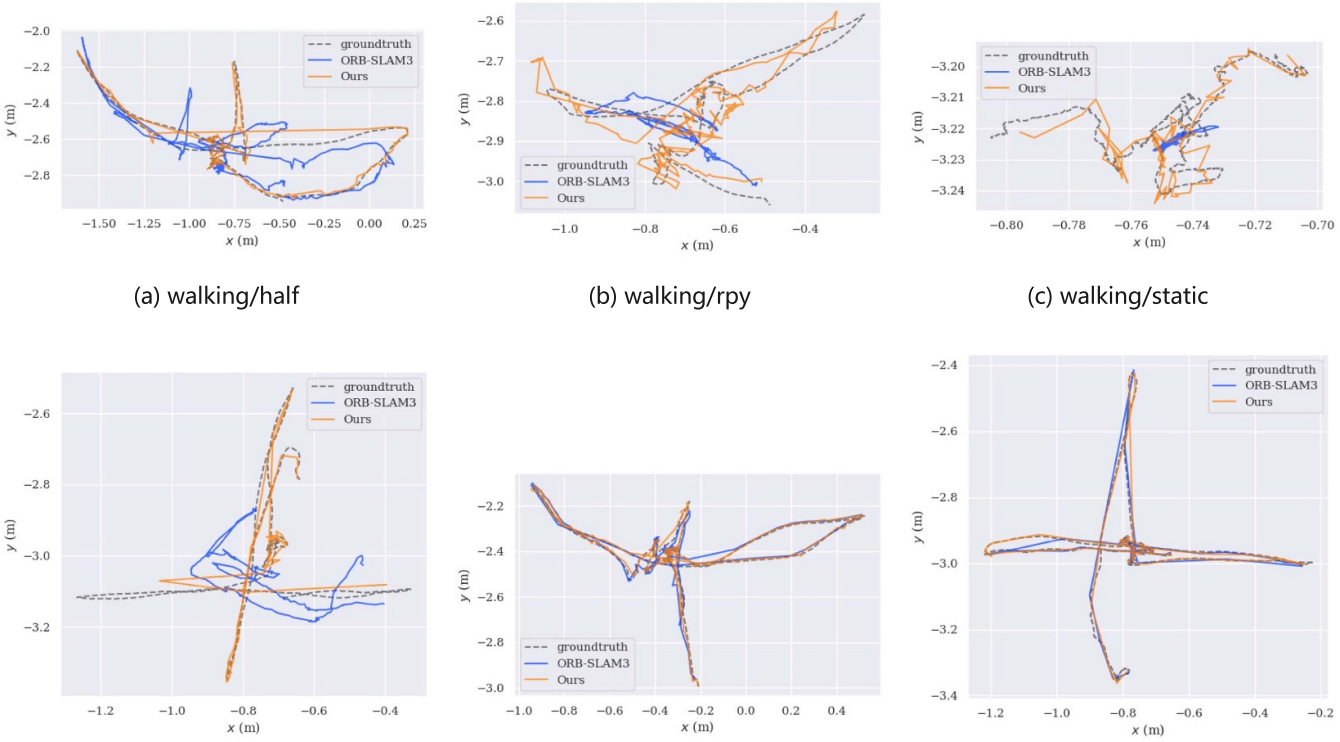
The third group includes the newly proposed SLAM system with good performance. Iccrf represents an advanced system [15] based on geometric methods, while other methods are based on semantics. Towards represents the method proposed by Ji *et al* [19], and Blitz represents Blitz-SLAM [22]. As presented in table 3, our SLAM shows the more accurate result in both low dynamic and high dynamic environments. Meanwhile, our system's ATE only in s/xyz and w/rpy is relatively low.

4.7.2. Bonn dataset. As illustrated in table 4, we conduct a thorough comparison of our method's mean ATE with the cutting-edge, dynamic vSLAM approaches, namely SG-SLAM [32] and Iccrf-SLAM. To ensure a more comprehensive evaluation, we also incorporate an advanced dense reconstruction method, MF [33], into our analysis.

In comparison with the baseline system, ORB-SLAM3, our proposed method exhibits a remarkable performance enhancement of over 90% in the crowd and person_tracking sequences, where human presence is a dominant feature of the scene. Additionally, the ATE is substantially diminished

Table 1. Compared with the SLAM systems based on ORB-SLAM3, in RMSE and S.D of the ATE and t.RPE. The best results are highlighted in bold, and we utilize the values in their original papers when available (m).

	ATE				t.RPE			
Sequence	O3 RMSE (S.D)	RDS RMSE (S.D)	DeepLab RMSE (S.D)	Ours RMSE (S.D)	O3 RMSE (S.D)	RDS RMSE (S.D)	DeepLab RMSE (S.D)	Ours RMSE (S.D)
w/half	0.193 (0.127)	0.025 (0.017)	0.027 (0.012)	**0.019 (0.010)**	0.022 (0.016)	0.027 (0.014)	0.023 (0.010)	**0.018 (0.009)**
w/rpy	0.137 (0.062)	0.146 (0.105)	**0.031 (0.018)**	0.053 (0.031)	**0.013 (0.009)**	0.024 (0.012)	0.040 (0.023)	0.035 (0.024)
w/static	0.019 (0.011)	0.081 (0.022)	0.006 **(0.002)**	**0.005 (0.002)**	**0.003 (0.002)**	0.022 (0.014)	0.008 (0.003)	0.006 (0.003)
w/xyz	0.235 (0.077)	0.021 (0.012)	**0.013 (0.006)**	**0.013** (0.008)	0.039 (0.030)	0.026 (0.016)	**0.017** (0.009)	**0.017 (0.011)**
s/half	0.020 (0.011)	—	—	**0.013 (0.005)**	**0.010 (0.005)**	—	—	0.012 (0.006)
s/xyz	0.012 (0.006)	—	—	**0.011 (0.005)**	0.016 (0.008)	—	—	**0.012 (0.005)**



(a) walking/half (b) walking/rpy (c) walking/static

(d) walking/xyz (e) sitting/half (f) sitting/xyz

Figure 5. The contrast of trajectories obtained from ORB-SLAM3 and our system against the real trajectory in TUM dataset.

Table 2. Compared with the SLAM systems referred to in our paper and system considered authoritative, in RMSE and S.D of the ATE and t.RPE. The best results are highlighted in bold, and we utilize the values in their original papers when available (m).

	ATE				t.RPE			
Sequence	Dyna RMSE (S.D)	DS RMSE (S.D)	Crowd RMSE (S.D)	Ours RMSE (S.D)	Dyna RMSE (S.D)	DS RMSE (S.D)	Crowd RMSE (S.D)	Ours RMSE (S.D)
w/half	0.029 (0.015)	0.030 (0.026)	0.026 (—)	**0.019 (0.010)**	0.028 (0.014)	0.030 (0.026)	0.037 (—)	**0.018 (0.009)**
w/rpy	0.035 (0.019)	**0.044** (0.377)	**0.044** (—)	0.053 **(0.031)**	0.044 (0.026)	0.150 (0.094)	0.065 (—)	**0.035 (0.024)**
w/static	0.006 (0.003)	0.008 (0.007)	0.007 (—)	**0.005 (0.002)**	0.008 (0.004)	0.010 (0.009)	0.010 (—)	**0.006 (0.003)**
w/xyz	0.016 (0.008)	0.024 (0.019)	0.020 (—)	**0.013 (0.008)**	0.021 **(0.011)**	0.033 (0.024)	0.025 (—)	**0.017 (0.011)**
s/half	0.018 (0.008)	—	0.020 (—)	**0.013 (0.005)**	0.023 (0.012)	—	0.022 (—)	**0.012 (0.006)**
s/xyz	0.012 (0.006)	—	0.018 (—)	**0.011 (0.005)**	0.014 (0.007)	—	0.020 (—)	**0.012 (0.005)**

in the balloon2 sequence and the placing_nonobstructing_box sequence, both of which are characterized by the presence of diverse object types.

Furthermore, when juxtaposed with other state-of-the-art dynamic vSLAM systems, our method demonstrates superior results in the majority of sequences (11/15). Notably,

Table 3. Compared with the latest SLAM systems, in RMSE and S.D of the ATE and t.RPE. The best results are highlighted in bold, and we utilize the values in their original papers when available (m).

	ATE				t.RPE			
Sequence	Towards RMSE (S.D)	Iccrf RMSE (S.D)	Blitz RMSE (S.D)	Ours RMSE (S.D)	Towards RMSE (S.D)	Iccrf RMSE (S.D)	Blitz RMSE (S.D)	Ours RMSE (S.D)
w/half	0.029 (—)	0.028 (0.015)	0.025 (0.012)	**0.019 (0.010)**	0.042 (—)	0.035 (0.024)	0.025 (0.012)	**0.018 (0.009)**
w/rpy	0.037 (—)	0.046 (0.034)	**0.035 (0.022)**	0.053 (0.031)	0.047 (—)	0.050 (0.046)	0.047 (0.028)	**0.035 (0.024)**
w/static	0.011 (—)	0.011 (0.008)	0.010 (0.005)	**0.005 (0.002)**	0.011 (—)	0.014 (0.011)	0.012 (0.006)	**0.006 (0.003)**
w/xyz	0.019 (—)	0.016 (0.011)	0.015 **(0.007)**	**0.013** (0.008)	0.023 (—)	0.021 (0.015)	0.019 (0.009)	**0.017 (0.011)**
s/half	0.017 (—)	—	0.016 (0.007)	**0.013 (0.005)**	0.025 (—)	—	0.016 (0.007)	**0.012 (0.006)**
s/xyz	0.011 (—)	**0.009 (0.005)**	0.014 (0.006)	0.011 **(0.005)**	0.016 (—)	**0.012** (0.007)	0.014 (0.007)	**0.012 (0.005)**

Table 4. Compared with the advanced vSLAM systems, in terms of the ATE's RMSE on the Bonn dataset, the best results are highlighted in bold, while the second-best results are underlined. We utilize the values reported in their original papers when available (m).

Sequence	ORB-SLAM3	SG-SLAM	MF	Iccrf	Ours
balloon	0.060	—	0.164	**0.027**	**0.027**
balloon2	0.141	—	0.114	0.024	**0.019**
balloon_tracking	0.031	—	0.194	**0.025**	0.029
balloon_tracking2	0.029	—	0.238	0.045	**0.022**
crowd	0.368	0.019	0.473	0.019	**0.016**
crowd2	0.507	0.042	0.653	0.031	**0.025**
crowd3	0.318	0.023	0.341	0.023	**0.019**
moving_nonobstructing_box2	0.031	0.028	0.193	0.038	**0.023**
person_tracking	0.591	0.038	0.301	**0.035**	0.040
person_tracking2	0.725	0.034	0.220	0.040	**0.032**
placing_nonobstructing_box	0.714	—	0.325	**0.014**	0.033
placing_nonobstructing_box2	0.026	—	0.153	**0.016**	0.020
placing_nonobstructing_box3	0.071	—	0.156	0.036	**0.021**
placing_obstructing_box	0.234	—	0.424	0.320	**0.109**
removing_nonobstructing_box	0.011	—	0.058	0.013	**0.009**

Table 5. Compared with the advanced vSLAM systems, in terms of the ATE's mean on the KITTI dataset, the best results are highlighted in bold, while the second-best results are underlined. We utilize the values reported in their original papers when available (m).

Sequence	ORB-SLAM3	DynaSLAM	DOT-SLAM	+IMU	Ours
00	**0.87**	1.4	1.18	0.91	**0.87**
01	10.60	9.4	**1.33**	2.08	9.26
02	3.06	6.7	1.04	**0.74**	3.57
03	0.33	0.6	1.00	1.72	**0.32**
04	0.18	0.2	1.00	1.81	**0.16**
05	0.37	0.8	1.14	—	**0.32**
06	0.42	0.8	1.07	**0.26**	0.38
07	**0.38**	0.5	1.00	2.26	0.42
08	2.52	3.5	**1.24**	—	2.65
09	1.03	1.6	3.89	2.09	**0.97**
10	1.51	1.2	**1.02**	2.39	1.29

our approach showcases a significant advantage in the three crowd sequences, marked by a high density of dynamic objects. Impressively, our system also performs well in the placing_obstructing_box sequence, wherein dynamic objects constitute approximately 80% of the scene. This compelling evidence reinforces the notion that our method thrives in complex scenarios featuring a rich assortment and a larger quantity of dynamic objects.

4.7.3. KITTI dataset. Table 5 presents the root mean square error (RMSE) of ATE for each system across the ten sequences. We compare our method with state-of-the-art outdoor environment approaches, such as DynaSLAM and DOT-SLAM [34], and show the effectiveness of our work by including a method with IMU integration. In the table, ORB-SLAM3 represents the baseline system that we have modified. DynaSLAM and DOT-SLAM denote two advanced outdoor

Table 6. Comparison of computation time (ms).

Method	Semantic part	Geometry part	Tracking
ORB-SLAM3	—	—	11.15
Ours	10.34	14.55	37.96

vSLAM systems, while '+IMU' signifies a cutting-edge VIO system employed in autonomous driving. 'Ours' refers to the system proposed in this paper.

Compared to the ORB-SLAM3 system, our proposed method has been demonstrated to reduce the absolute trajectory error. The empirical evaluation reveals that the trajectory error is reduced by approximately 10%–15% in the majority of the sequences. Notably, our method yields unsatisfactory results in three sequences recorded in rural settings (02, 07, and 08). These sequences exhibit high traffic density and close distance between stationary and moving vehicles. Consequently, our system tends to erroneously remove points from stationary vehicles, resulting in degraded performance. However, our system performs well in sequences 01 and 04, captured on spacious highways, and other sequences recorded in urban and rural areas with fewer parked vehicles. This indicates that our system can effectively handle multiple moving vehicles on highways.

Compared with the IMU-enhanced method, our approach shows a significant gap in sequences 01 and 02, which feature complex lighting conditions. In the remaining sequences, apart from sequence 6, our method achieved more than a 50% reduction in ATE. This demonstrates the superiority of our purely visual approach. Furthermore, when compared to DOT-SLAM and Dyna-SLAM, our method yields better results in most sequences, except for sequences 01 and 10.

4.8. Runtime analysis

In practical applications, the ability of a system to operate in real-time is of paramount importance. Given the TUM dataset's reputation as a highly regarded and widely employed benchmark within the field, we have chosen it as the basis for our runtime analysis. This decision facilitates a rigorous and dependable assessment of our approach within a well-established framework. To gauge the real-time performance of our SLAM system, we calculate the average processing time for each image across different sub-modules. Concurrently, we compare the tracking time of the baseline ORB-SLAM3 and our proposed method, further emphasizing the significance of real-time functionality in our research.

As shown in table 6, there are three different modules with processing time in milliseconds. The tracking time of this system is only 26 ms higher than that of the original ORB-SLAM3. In addition, the processing time of the Semantic Module accelerated by TensorRT reaches 10.34 ms per frame, which is 4.21 ms faster than the geometry module. In a word, our system meets the real-time[3] requirement.

[3] Real-time in our paper refers to the time of processing images by the robot is the same as humanity's brain's, i.e. 100 ms per frame [35].

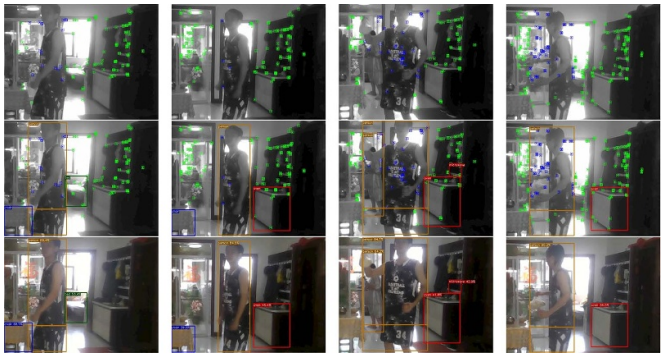


Figure 6. Experiment with a monocular camera in a real-life scene. Our method successfully detects dynamic points belonging to known objects (people) and limited unknown moving objects (plastic bags).

4.9. Robustness test in real environment

We evaluate our method in real-world environments by utilizing the monocular camera of a laptop, which not only showcases the efficacy of our approach in handling real dynamic scenarios but also demonstrates the universality of our method. During the experimentation, we manually shook and rotated the camera while moving it. Simultaneously, one individual performed various actions in front of the camera, while another person unknowingly entered the camera's field of view and engaged in random movements. Figure 6 illustrates the specific results of the moving point culling algorithm. The first row displays the detected objects within the image, the second row presents the discernment of moving and static boxes, and the third row exhibits the dynamic points in blue and the static points in green.

The first column highlights the dynamic points detected on the human body, while the second column showcases the effectiveness of the NHBC algorithm. In the third column, a woman inadvertently enters the camera frame while retrieving something from the refrigerator, with the dynamic points on her body accurately marked in blue. The fourth column accurately identifies the dynamic key points on the undetected plastic bags, marking only the points on the moving upper human body as dynamic, while the static lower body is correctly classified as static.

In the absence of ground truth trajectories in real-world scenarios, we evaluate the influence of small displacements and jitters within indoor settings. The presence of dynamic objects can induce considerable deviations in the estimated trajectory length. As depicted in figure 7, there is an approximate five-fold disparity between the scales of the trajectories estimated by ORB-SLAM3 (with the x-axis in units of 0.1 m) and our method (with the x-axis in units of 0.02 m).

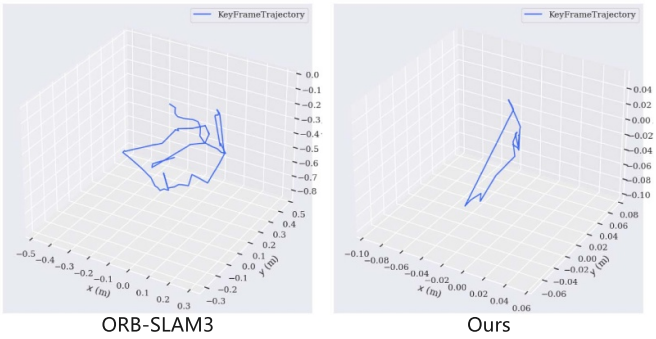


Figure 7. The contrast of trajectories obtained from ORB-SLAM3 and our system in real environment.

This observation suggests that minor movements have been erroneously amplified into more substantial displacements. Furthermore, a trajectory that should have resembled a small elliptical closed loop is estimated by ORB-SLAM3 as a circular path with extraneous trajectories outside the closed loop. This evidence further underscores the notable deviation manifested by ORB-SLAM3 and the robustness of our approach under monocular conditions, emphasizing the efficacy of our method in tackling the challenges presented by dynamic environments.

5. Conclusion

We propose a real-time vSLAM system capable of handling both known and limited unknown dynamic objects in dynamic environments. Our approach emphasizes retaining a greater number of static boxes to increase the number of stable key points while mitigating the influence of dynamic objects. Through experimental evaluation, our method demonstrates superior accuracy compared to other advanced techniques and exhibits commendable real-time performance. The robustness of our approach is substantiated by various indoor scenarios in the TUM and Bonn datasets, while experiments on the KITTI dataset establish its effective operation in outdoor environments under the stereo camera. Moreover, real-world monocular camera tests reveal that our system maintains good performance even under monocular settings. Additionally, our method can detect a higher number of semantic objects than other approaches, indicating its potential to better construct long-term semantic maps in dynamic environments for accomplishing advanced robotic tasks [36–38]. Overall, our system represents the only known by us universal vSLAM system capable of performing well across all types of cameras and environments.

During the experimental process, we attempted to divide humans into four parts and objects into two parts based on scale or solely eliminate key points belonging to humans; however, these modifications resulted in reduced system performance. Simultaneously, although the NHBC is not robust for local human body parts from different angles, it imposes virtually zero burdens on the system. In the future, boxes can be evaluated using additional information, such as reprojection errors, to facilitate more comprehensive judgment and further enhance system performance.

Data availability statements

The data that support the findings of this study are available at the following URL

https://vision.in.tum.de/data/datasets/rgbd-dataset/download [29].

https://www.ipb.uni-bonn.de/html/projects/rgbd_dynamic2019/rgbd_bonn_dataset.zip [30].

https://s3.eu-central-1.amazonaws.com/avg-kitti/data_odometry_color.zip [31].

The data that support the findings of this study are available upon reasonable request from the authors.

ORCID iD

Changdi Li https://orcid.org/0000-0002-3638-7249

References

[1] Li C, Yu L and Fei S 2020 Large-scale, real-time 3D scene reconstruction using visual and IMU sensors *IEEE Sens. J.* **20** 5597–605

[2] Li C, Yu L and Fei S 2019 Real-time 3D motion tracking and reconstruction system using camera and IMU sensors *IEEE Sens. J.* **19** 6460–6

[3] Li Y and Yan K 2021 Indoor localization based on radio and sensor measurements *IEEE Sens. J.* **21** 25090–7

[4] Ren G, Cao Z, Liu X, Tan M and Yu J 2022 PLJ-SLAM: monocular visual SLAM with points, lines and junctions of coplanar lines *IEEE Sens. J.* **22** 15465–76

[5] Campos C, Elvira R, Rodríguez J J G, Montiel J M M and Tardós J D 2021 ORB-SLAM3: an accurate open-source library for visual, visual–inertial and multimap SLAM *IEEE Trans. Robot.* **37** 1874–90

[6] Chien C-H, Hsu C-C J, Wang W-Y and Chiang H-H 2020 Indirect visual simultaneous localization and mapping based on linear models *IEEE Sens. J.* **20** 2738–47

[7] Fischler M A and Bolles R C 1981 Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography *Commun. ACM* **24** 381–95

[8] Soares J C V, Gattass M and Meggiolaro M A 2021 Crowd-SLAM: visual SLAM towards crowded environments using object detection *J. Intell. Robot. Syst.* **102** 50

[9] Badrinarayanan V, Kendall A and Cipolla R 2017 SegNet: a deep convolutional encoder-decoder architecture for image segmentation *IEEE Trans. Pattern Anal. Mach. Intell.* **39** 2481–95

[10] He K, Gkioxari G, Dollár P and Girshick R 2017 Mask R-CNN *2017 IEEE Int. Conf. on Computer Vision (ICCV)* pp 2980–8

[11] Ge Z, Liu S, Wang F, Li Z and Sun J 2021 YOLOX: exceeding YOLO series in 2021 (arXiv:2107.08430)

[12] Sun Y, Liu M and Meng M Q-H 2017 Improving RGB-D SLAM in dynamic environments: a motion removal approach *Robot. Auton. Syst.* **89** 110–22

[13] Zhang T, Zhang H, Li Y, Nakamura Y and Zhang L 2020 FlowFusion: dynamic dense RGB-D SLAM based on optical flow (arXiv:2003.05102)

[14] Dai W, Zhang Y, Li P, Fang Z and Scherer S 2022 RGB-D SLAM in dynamic environments using point correlations *IEEE Trans. Pattern Anal. Mach. Intell.* **44** 373–89
[15] Scona R, Jaimez M, Petillot Y R, Fallon M and Cremers D 2018 StaticFusion: background reconstruction for dense RGB-D SLAM in dynamic environments *2018 IEEE Int. Conf. on Robotics and Automation (ICRA)* pp 3849–56
[16] Du Z-J, Huang S-S, Mu T-J, Zhao Q, Martin R R and Xu K 2022 Accurate dynamic SLAM using CRF-based long-term consistency *IEEE Trans. Vis. Comput. Graph.* **28** 1745–57
[17] Bescos B, Fácil J M, Civera J and Neira J 2018 DynaSLAM: tracking, mapping and inpainting in dynamic scenes *IEEE Robot. Autom. Lett.* **3** 4076–83
[18] Redmon J and Farhadi A 2018 YOLOv3: an incremental improvement (arXiv:1804.02767)
[19] Ji T, Wang C and Xie L 2021 Towards real-time semantic RGB-D SLAM in dynamic environments *2021 IEEE Int. Conf. on Robotics and Automation (ICRA)* pp 11175–81
[20] Yu C, Liu Z, Liu X-J, Xie F, Yang Y, Wei Q and Fei Q 2018 DS-SLAM: a semantic visual SLAM towards dynamic environments *2018 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)* pp 1168–74
[21] Cheng J, Wang Z, Zhou H, Li L and Yao J 2020 DM-SLAM: a feature-based SLAM system for rigid dynamic scenes *ISPRS Int. J. Geo-Inf.* **9** 202
[22] Fan Y, Zhang Q, Tang Y, Liu S and Han H 2022 Blitz-SLAM: a semantic SLAM in dynamic environments *Pattern Recognit.* **121** 108225
[23] Dvornik N, Shmelkov K, Mairal J and Schmid C 2017 BlitzNet: a real-time deep network for scene understanding *2017 IEEE Int. Conf. on Computer Vision (ICCV)* pp 4174–82
[24] Hu Z, Zhao J, Luo Y and Ou J 2022 Semantic SLAM based on improved DeepLabv3+ in dynamic scenarios *IEEE Access* **10** 21160–8
[25] Chen L-C, Zhu Y, Papandreou G, Schroff F and Adam H 2018 Encoder-decoder with atrous separable convolution for semantic image segmentation *Computer Vision—ECCV 2018*, ed V Ferrari, M Hebert, C Sminchisescu and Y Weiss (Cham: Springer) pp 833–51
[26] Liu Y and Miura J 2021 RDS-SLAM: real-time dynamic SLAM using semantic segmentation methods *IEEE Access* **9** 23772–85
[27] Nvidia (available at: https://developer.nvidia.com/tensorrt)
[28] Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P and Zitnick C L 2014 Microsoft COCO: common objects in context *Computer Vision—ECCV 2014*, ed D Fleet, T Pajdla, B Schiele and T Tuytelaars (Cham: Springer) pp 740–55
[29] Sturm J, Engelhard N, Endres F, Burgard W and Cremers D 2012 A benchmark for the evaluation of RGB-D SLAM systems *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* pp 573–80
[30] Palazzolo E, Behley J, Lottes P, Giguère P and Stachniss C 2019 ReFusion: 3D reconstruction in dynamic environments for RGB-D cameras exploiting residuals *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)* pp 7855–62
[31] Geiger A, Lenz P and Urtasun R 2012 Are we ready for autonomous driving? the kitti vision benchmark suite *Conf. on Computer Vision and Pattern Recognition (CVPR)*
[32] Cheng S, Sun C, Zhang S and Zhang D 2023 SG-SLAM: a real-time RGB-D visual SLAM toward dynamic scenes with semantic and geometric information *IEEE Trans. Instrum. Meas.* **72** 1–12
[33] Runz M, Buffier M and Agapito L 2018 MaskFusion: real-time recognition, tracking and reconstruction of multiple moving objects *2018 IEEE Int. Symp. on Mixed and Augmented Reality (ISMAR)* pp 10–20
[34] Ballester I, Fontán A, Civera J, Strobl K H and Triebel R 2021 DOT: dynamic object tracking for visual SLAM *2021 IEEE Int. Conf. on Robotics and Automation (ICRA)* (IEEE Press) pp 11705–11
[35] Potter M C and Levy E I 1969 Recognition memory for a rapid sequence of pictures *J. Exp. Psychol.* **81** 10
[36] Hou J, Yu L, Li C and Fei S 2020 Handheld 3D reconstruction based on closed-loop detection and nonlinear optimization *Meas. Sci. Technol.* **31** 025401
[37] Cheng J, Wang C, Mai X, Min Z and Meng M Q-H 2021 Improving dense mapping for mobile robots in dynamic environments based on semantic information *IEEE Sens. J.* **21** 11740–7
[38] Lv Z, Mauri J L and Song H 2020 Editorial RGB-D sensors and 3D reconstruction *IEEE Sens. J.* **20** 11751–2