ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/374024828

MVS-SLAM: Enhanced multiview geometry for improved semantic RGBD SLAM in dynamic environment

Article *in* Journal of Field Robotics · September 2023

DOI: 10.1002/rob.22248

CITATIONS
0

READS
146

5 authors, including:

Qamar Ul Islam
Universiti Sains Malaysia
39 PUBLICATIONS **26** CITATIONS

SEE PROFILE

Haidi Ibrahim
Universiti Sains Malaysia
119 PUBLICATIONS **3,309** CITATIONS

SEE PROFILE

Mohd Zaid Abdullah
Universiti Sains Malaysia
75 PUBLICATIONS **816** CITATIONS

SEE PROFILE

All content following this page was uploaded by Qamar Ul Islam on 21 September 2023.

The user has requested enhancement of the downloaded file.

Received: 5 April 2023 | Revised: 18 August 2023 | Accepted: 4 September 2023

DOI: 10.1002/rob.22248

RESEARCH ARTICLE

WILEY

MVS-SLAM: Enhanced multiview geometry for improved semantic RGBD SLAM in dynamic environment

Qamar Ul Islam[1] | Haidi Ibrahim[1] | Pan Kok Chin[2] | Kevin Lim[2] | Mohd Zaid Abdullah[1]

[1]School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Engineering Campus, Nibong Tebal, Penang, Malaysia

[2]PixArt Imaging (Penang), Sdn. Bhd., Kompleks Eureka, Universiti Sains Malaysia, Gelugor, Penang, Malaysia

Correspondence
Mohd Zaid Abdullah, School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Engineering Campus, 14300 Nibong Tebal, Penang, Malaysia.
Email: mza@usm.my

Funding information
Pixart Imaging (6050393/P153); Universiti Sains Malaysia (8070007)

Abstract

Simultaneous Localization and Mapping (SLAM) is a crucial technology for intelligent mobile robots to operate successfully in unknown environments. While many excellent SLAM systems have been developed in recent years, most assume that the environment is static, resulting in poor performance in dynamic environments. To address this limitation, we propose multiview stereo (MVS)-SLAM (MVS-SLAM), a real-time semantic RGBD SLAM system with improved-multiview geometry, built on the RGBD mode of ORB-SLAM3. MVS-SLAM tightly integrates semantic and geometric information to tackle the challenges posed by dynamic scenes. To meet the real-time requirements, the semantic module leverages the latest and fastest object detection network, YOLOv7, to provide semantic prior knowledge for the geometric module. We introduce a novel geometric constraint method that capitalizes on depth images and semantic information to recover three-dimensional (3D) feature points and initial camera pose. We use a 3D coordinate error threshold to identify dynamic points and remove them using the *K*-means clustering algorithm. This approach effectively reduces the impact of dynamic points. We validate MVS-SLAM using challenging dynamic sequences from the TUM data set, demonstrating that it significantly improves localization accuracy and system robustness in all types of dynamic environments.

KEYWORDS
improved-multiview geometry, object detection network, optical flow, RGBD SLAM

1 | INTRODUCTION

In recent years, there has been a significant advancement in autonomous robots, augmented reality, and unmanned aerial vehicle technologies (Fang et al., 2021; Li et al., 2017; Zhao et al., 2021). Particularly important tasks include reconstructing the surrounding environment (Endres et al., 2013) and estimating the robot's ego state (Davison, 2003). Among the various types of visual cameras, including monocular, stereo, and RGBD cameras, RGBD cameras have gained attention due to their low cost and ability to provide rich image information attention due to their low cost and ability to provide rich image information. They serve as the "eyes" of the robot, enabling observation of the environment. In the field of Visual Simultaneous Localization and Mapping, numerous Simultaneous Localization and Mapping (SLAM) systems have emerged, including ORB-SLAM3 (Campos et al., 2021), semidirect visual odometry (Forster et al., 2014), Vins-mono (Qin et al., 2018), and LSD-SLAM (Engel et al., 2014), which have demonstrated satisfactory performance in static environments.

However, real-world environments are not static but instead contain dynamic objects that can introduce errors in camera motion estimation. Feature points associated with dynamic objects can lead

 © 2023 Wiley Periodicals LLC.

to incorrect matches and disrupt the tracking process. Although traditional SLAM systems employ techniques such as Random Sample Consensus (RANSAC) (Fischler & Bolles, 1981) and robust cost functions to mitigate the impact of dynamic points, their effectiveness diminishes when faced with a large number of dynamic objects or when dynamic objects dominate the scene. In recent years, deep neural networks (DNNs) have made significant contributions to computer vision research (Zhao et al., 2019). DNNs have been successfully applied to various tasks, such as object detection, semantic segmentation, motion tracking, three-dimensional (3D) reconstruction, and action recognition. Recognizing the potential benefits of integrating DNNs into visual SLAM, researchers have begun exploring the integration of semantic information provided by DNNs to mitigate the negative effects of dynamic points.

In this paper, we propose a real-time semantic SLAM system called multiview stereo (MVS)-SLAM (MVS-SLAM), which builds upon the foundation of ORB-SLAM3 (Campos et al., 2021). To address the challenge of dynamic points, we introduce three key techniques. First, an object detection module is integrated into the ORB-SLAM3 system as a separate thread to provide semantic prior knowledge for the selection of feature points. Second, an ego-motion estimation module utilizes semantic prior information to identify static feature points and recover the initial camera pose. Lastly, a dynamic feature points removal module leverages the spatial coordinate error as a threshold to eliminate pure dynamic feature points, allowing only static feature points to be used for mapping and tracking within the SLAM system.

The primary contributions of this paper lie in the proposed techniques that effectively mitigate the impact of dynamic points on SLAM performance. Additionally, our approach incorporates improved-multiview geometry, enhancing the accuracy of camera pose estimation and 3D reconstruction. Overall, this paper introduces several valuable contributions to the field of visual SLAM. we present improved-multiview geometry in our approach, which enhances the accuracy of camera pose estimation and 3D reconstruction. The paper proposes several contributions to the field of visual SLAM:

- *Integration of YOLOv7 object detection network*: The paper incorporates the YOLOv7 object detection network into the ORB-SLAM3 system, enabling the efficient and accurate extraction of semantic prior information. This integration enhances the system's capability to quickly identify and utilize relevant objects in the environment.
- *Ego-motion estimation module*: The proposed ego-motion estimation module introduces two strategies to improve robustness and speed in recovering the initial camera pose. By leveraging semantic prior information, the module enhances the accuracy of estimating camera motion, leading to more reliable localization results.
- *Novel geometric constraint method*: The paper introduces a novel geometric constraint method that effectively removes pure dynamic points while preserving static points. This technique enables the system to filter out feature points associated with dynamic objects, minimizing the impact of dynamic environments on the SLAM performance.
- *Real-time improved-multiview geometry semantic visual SLAM system*: The presented system is a real-time solution designed specifically for indoor dynamic environments. By incorporating improved-multiview geometry techniques, the system achieves enhanced localization accuracy. Experimental evaluations on the TUM data set, including both high- and low-dynamic sequences, demonstrate the effectiveness of the proposed approach.

In summary, the contributions of this paper encompass the integration of YOLOv7 object detection, the development of an ego-motion estimation module, the introduction of a novel geometric constraint method, and the implementation of a real-time improved-multiview geometry semantic visual SLAM system. These contributions collectively enhance the system's performance in dynamic environments, leading to improved localization accuracy and demonstrating the effectiveness of the proposed approach.

The paper is organized as follows: In Section 2, an overview of related works is presented. Section 3 describes the proposed MVS-SLAM method in detail. Section 4 presents the experimental results obtained from evaluating the proposed method on the TUM data set, and compares its performance with other state-of-the-art SLAM systems, such as DS-SLAM and Dynamic SLAM (DynaSLAM). Finally, Section 5 provides a brief conclusion and outlines possible directions for future work.

2 | RELATED WORK

In this section, we provide an extensive review of the existing literature related to SLAM, RGBD sensing, and semantic understanding in dynamic environments. We discuss key approaches, methodologies, and limitations of the state-of-the-art methods, aiming to establish a solid foundation for understanding the gaps in current research and the need for our proposed MVS-SLAM approach. In dynamic situations, traditional SLAM systems will face significant problems. Several scholars are concerned with reducing the influence of dynamic points. There are three types of major approaches: geometry methods, deep learning methods, and combinations of geometry and deep learning methods. In this part, we will look at many SLAM systems that use these three strategies.

2.1 | SLAM in dynamic environments

SLAM techniques have made significant advancements in recent years, enabling robots, and autonomous systems to navigate and map their surroundings in real-time. Traditional SLAM approaches primarily focused on static environments, assuming static scenes and camera motion. However, dynamic environments pose several challenges due to moving objects, changing appearances, and occlusions. To address these challenges, researchers have proposed various methods to improve SLAM performance in dynamic environments. One popular approach is DynaSLAM by Bescos et al. (2018),

which incorporates dynamic object detection and tracking to separate moving objects from the static scene. DynaSLAM achieves accurate camera pose estimation by considering the static background while mitigating the influence of moving objects. However, it still relies on rigid motion assumptions and struggles with complex scene dynamics. Another notable method is DM-SLAM, as presented by Cheng, Wang, Zhou et al. (2020). DM-SLAM leverages deep learning techniques to enhance SLAM performance in dynamic scenes. By incorporating a dynamic object detection network, DM-SLAM (Cheng, Wang, Zhou, et al., 2020) can accurately track camera poses and separate dynamic objects from the static scene. However, its reliance on deep learning models introduces computational overhead and limits its generalizability to various platforms and sensor configurations.

2.2 | RGBD sensing and semantic understanding

RGBD sensors, such as Microsoft Kinect and Intel RealSense, have become increasingly popular in robotic applications, providing depth information alongside RGB imagery. These additional depth data enable the extraction of 3D information and facilitate accurate scene reconstruction and object detection. ORB-SLAM2 (Mur-Artal & Tardós, 2017) is a widely used RGBD SLAM system that combines feature-based tracking and mapping with loop closure detection. It achieves robust localization and mapping by exploiting depth information and features extracted from RGB images. However, ORB-SLAM2 primarily focuses on geometric understanding and lacks explicit semantic understanding of the environment. ORB-SLAM3 (Campos et al., 2021), an extension of ORB-SLAM2, addresses this limitation by incorporating semantic information into the SLAM framework. By leveraging semantic SegNets, ORB-SLAM3 can associate semantics with the reconstructed 3D map, enabling a higher-level understanding of the scene. However, ORB-SLAM3's semantic understanding capabilities are limited to static scenes and do not account for dynamic objects.

2.3 | Geometry-based methods

The primary goal of the geometry technique is to distinguish between dynamic and static locations using geometric differences. To construct 3D feature points, Kim and Kim (2016) proposed merging RGB color information with depth information. Depending on the dynamically shifting properties, dynamic objects are classed as inliers or outliers. RANSAC is used to reject outliers. Wang et al. (2019) employed the fundamental matrix to identify feature point inconsistency and first grouped the depth picture into various groups before filtering outliers using the fundamental matrix. Lastly, based on the statistical features collected above, a moving objects judgment model is created. When the number of outliners in the cluster region exceeds a certain threshold, all features on it are removed. Dai et al. (2020) developed Delaunay triangulation and evaluated changes in triangle edges to discriminate between dynamic and static sites. With RGB pictures as the only input, Cheng, Wang, and Meng (2020) employed optical flow to differentiate and delete dynamic feature points from retrieved ones. Geometry approaches often employ strict mathematical functions and make use of pixel-level data. Geometry approaches often outperform deep learning methods in terms of speed. They lack high-level information, however, and cannot use past knowledge to comprehend the scenario. As a result, geometry approaches are less resilient than deep learning methods. Geometry approaches perform poorly, particularly when dynamic objects take up a major section of the picture.

2.4 | Deep learning-based methods

The basic concept behind the deep learning approach is to produce semantic prior information using cutting-edge deep learning networks. People, cars, and other highly dynamic things are identified as such based on the human experience. Afterward, any feature points with high-dynamic labels associated will be removed. Two kinds of concepts are typically used: semantic segmentation and object detection. Zhong et al. (2018) employed an object detection network SSD to recognize moving items, such as humans, dogs, cats, and cars. YOLO was used by Zhang et al. (2020) to get semantic messages. Li et al. (2020) utilized SegNet, a semantic segmentation network, to segment images. Deep learning networks aid SLAM systems in understanding their surroundings on a semantic level. Nevertheless, it merely gives semantic information, demonstrating the object's motion probability rather than its present state of motion. As a result, the deep learning algorithm cannot determine whether the item is moving or not at that time.

2.5 | Combination of geometry and deep learning-based methods learning-based methods

This technique combines the advantages of geometry with deep learning technologies. By studying the image in advance, the deep learning approach creates semantic prior knowledge for the subsequent phase. This semantic information is used by the geometry technique to accurately and effectively filter pure dynamic points. Yu et al. (2018) presented DS-SLAM, a semantic SLAM system that filters dynamic points using semantic segmentation with SegNet and epipolar restrictions. DS-SLAM also includes a real-time semantic segmentation thread to boost system performance. DynaSLAM, suggested by Bescos et al. (2018), combines multiview geometry with Mask regions with convolutional neural networks (R-CNN) to detect dynamic points in the RGBD scenario. Cui and Ma (2019) developed a closely connected approach termed semantic optical flow that removes dynamic feature points by utilizing feature point dynamic features buried in semantic and geometric information. Cheng, Wang, Zhou et al. (2020) introduced DM-SLAM, a four-module system that includes semantic segmentation, ego-motion estimation, dynamic

point detection, and a feature-based SLAM framework. To differentiate dynamic points, the dynamic point detection module employs reprojection offset vectors and epipolar restrictions. To differentiate dynamic characteristics in the detecting regions, Wu et al. (2022) used the Dark-net19-YOLOv3 network and depth difference using RANSAC. Lastly, You et al. (2022) removed pure dynamic feature points using YOLOCT++ instance segmentation and the error of reprojection depth as a threshold.

2.6 | Limitations and gaps

While the aforementioned methods have made significant contributions to SLAM and RGBD sensing in dynamic environments, there are still several limitations and gaps in the current state-of-the-art. First, most existing approaches assume rigid scene motion and struggle to handle complex dynamic scenes with nonrigid object deformations. Tracking and estimating the pose of nonrigidly moving objects remains a challenging problem in DynaSLAM.

Second, the semantic understanding (Garcia-Garcia et al., 2018) capabilities of current methods are limited, often restricted to static scenes or lacking explicit handling of dynamic objects. Achieving a comprehensive understanding of both static and dynamic elements in the scene is crucial for many robotic (Liu et al., 2021) applications. Third, the computational efficiency of existing methods varies, and some approaches rely on computationally expensive deep learning models, limiting their deployment on resource-constrained platforms. Overall, these limitations and gaps in the state-of-the-art methods highlight the need for a more comprehensive and efficient approach for SLAM in dynamic environments. In the following sections, we present our proposed MVS-SLAM approach, which aims to address these challenges by leveraging enhanced multiview geometry, dynamic object tracking, and semantic understanding.

In a traditional SLAM system (Tsai, 2012), the assumption of a static environment is often made, which limits their performance in dynamic scenarios. Our proposed MVS-SLAM system addresses this limitation by leveraging semantic information obtained from RGBD images. By utilizing the latest and fastest object detection network, YOLOv7 (Wang et al., 2023), we can extract real-time semantic information, which complements the geometric module of MVS-SLAM. The semantic module plays a crucial role in our system by providing valuable prior knowledge about the environment. This knowledge helps differentiate between static and dynamic elements present in the scene. By identifying and categorizing objects in real-time, the semantic module enhances the overall performance of MVS-SLAM. It assists in more reliable feature point recovery and camera pose estimation, leading to improved localization accuracy and robustness in dynamic environments.

By tightly integrating semantic and geometric information, MVS-SLAM tackles the challenges posed by dynamic scenes. The semantic information guides the geometric module in making more informed decisions, effectively reducing the impact of dynamic points and improving the overall consistency of the SLAM process. This integration enables our system to adapt and perform well in a wide range of dynamic environments, where traditional SLAM methods struggle. The significance of semantic information in MVS-SLAM lies in its ability to provide crucial contextual cues that aid in better understanding and modeling of the environment. The integration of semantic information allows for a more comprehensive understanding of the scene, resulting in improved scene reconstruction, mapping accuracy, and robustness against dynamic changes.

In general, the combination of geometry with deep learning is seen as a superior technique. Yet, there are still certain issues with present SLAM systems. Semantic segmentation has better precision but takes longer. This may result in the SLAM system failing to fulfill real-time requirements in actual applications. On the other hand, object detection is faster, but its bounding box may contain some static feature points. Additionally, some DynaSLAM algorithms erroneously eliminate all feature points in the dynamic region. As seen in Figure 12, a seated guy moves just his hands and head, while the rest of his body remains still. As a result, feature points placed in the static section of his body should not be eliminated; and certain DynaSLAM systems work well in high-dynamic sequences but poorly in low-dynamic sequences. In this study, we present MVS-SLAM, which may successfully tackle these challenges.

2.7 | Theoretical foundations

In this section, we provide a comprehensive overview of the theoretical principles that underpin our proposed MVS-SLAM approach. Understanding these foundations is essential for grasping the significance and technical aspects of our method. We discuss the key concepts of RGBD SLAM, semantic segmentation, multiview geometry, and their integration within our framework.

2.7.1 | RGBD SLAM

RGBD SLAM involves simultaneously estimating the camera pose and reconstructing the 3D environment using RGB and depth information. Our approach builds upon the widely used ORB-SLAM3 framework, which combines feature-based tracking, local mapping, and loop closing. We describe the theoretical foundations of ORB-SLAM3, including the oriented fast and rotated brief (ORB) feature detector, keypoint matching algorithms, and the Bundle Adjustment optimization technique. This understanding forms the basis for our subsequent enhancements.

2.7.2 | Semantic segmentation

To incorporate semantic information, we employ YOLOv7, a state-of-the-art deep-learning model for object detection and semantic segmentation. We delve into the theoretical principles of YOLOv7, including its architecture, training process, and inference mechanism. We emphasize how YOLOv7 leverages CNNs and anchor-based

object detection to assign semantic labels to the detected objects in the RGBD images.

2.7.3 | Multiview geometry

Multiview geometry plays a crucial role in our MVS-SLAM approach. We explore the theoretical foundations of multiview geometry, including epipolar geometries, camera projection models, and triangulation techniques. These principles enable us to establish correspondences between multiple views, estimate camera poses, and reconstruct the 3D scene. We also discuss how semantic information can be effectively fused with geometric information to enhance the accuracy and robustness of the SLAM system.

2.7.4 | Integration of semantic and geometric information

One of the core contributions of our approach is the seamless integration of semantic and geometric information. We elaborate on the theoretical framework that facilitates this integration, highlighting how semantic segmentation results are utilized to improve cam-era pose estimation, loop closure detection, and map refinement. We discuss the theoretical implications of incorporating semantic information within the SLAM pipeline, including its potential benefits for scene understanding, object tracking, and semantic mapping.

3 | SYSTEM DESCRIPTION

MVS-SLAM is based on ORB-SLAM3, which has three concurrent threads: Tracking, LocalMapping, and LocalClosing. ORB-SLAM3 is well known for its excellent performance in static environments, but it suffers in dynamic environments because dynamic feature points might generate mistakes in camera trajectory estimation. MVS-SLAM includes three new processes to solve this: a real-time object identification thread, an ego-motion estimate module, and a dynamic feature point removal module. Moreover, MVS-SLAM improves on ORB-multiview SLAM3's geometry technique, which includes triangulating feature points across several views to estimate camera postures. MVS-enhanced SLAM's multiview geometry considers dynamic objects and eliminates their input to the triangulation process, resulting in more accurate camera trajectory estimation.

3.1 | Overview of MVS-SLAM

Figure 1 depicts an overview of our proposed MVS-SLAM, while Figure 2 depicts more features in the pipeline. RGB and depth pictures

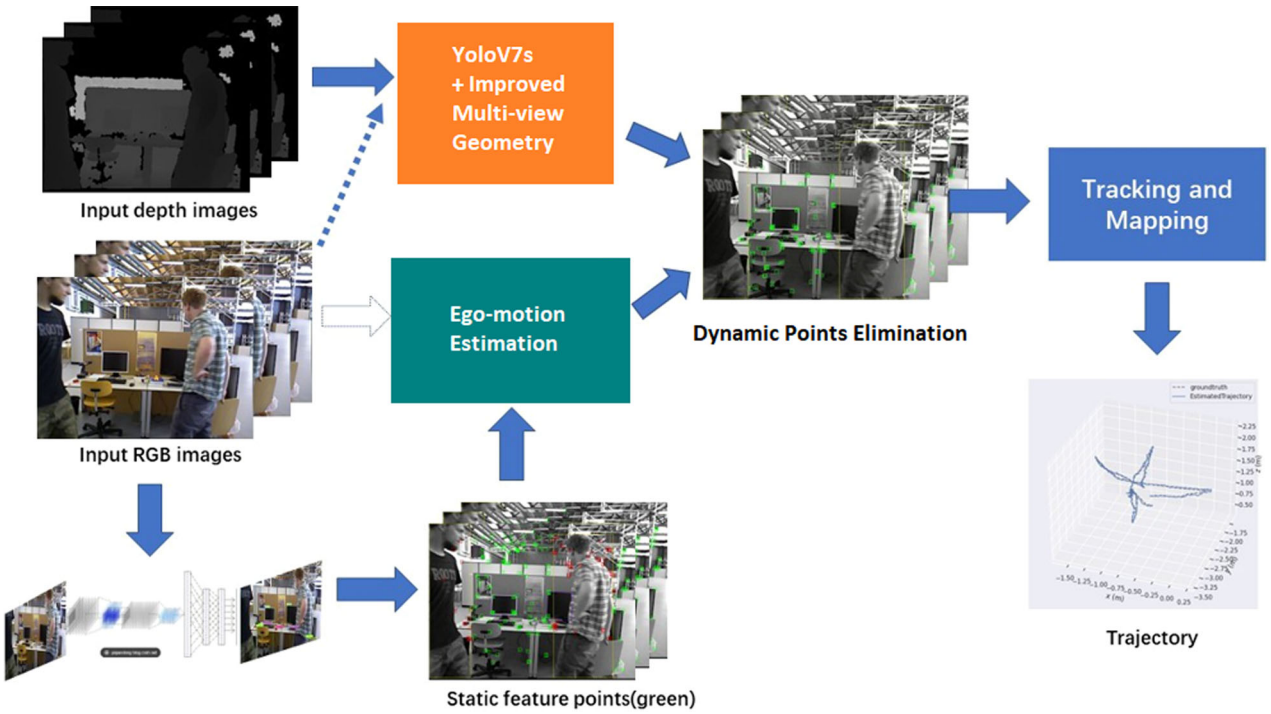

FIGURE 1 Schematic representation of the MVS-SLAM framework incorporating improved-multiview geometry. The YOLOv7 network performs object detection on RGB images, while static feature points are utilized to estimate the initial camera pose. The combination of RGB and depth images enables the recovery of spatial coordinates. The dynamic feature points removal module effectively distinguishes between dynamic and static feature points using reprojection offset vectors and subsequently removes pure dynamic points. The remaining static feature points are then utilized for tracking and mapping within the SLAM algorithm. MVS, multiview stereo; SLAM, Simultaneous Localization and Mapping. [Color figure can be viewed at wileyonlinelibrary.com]

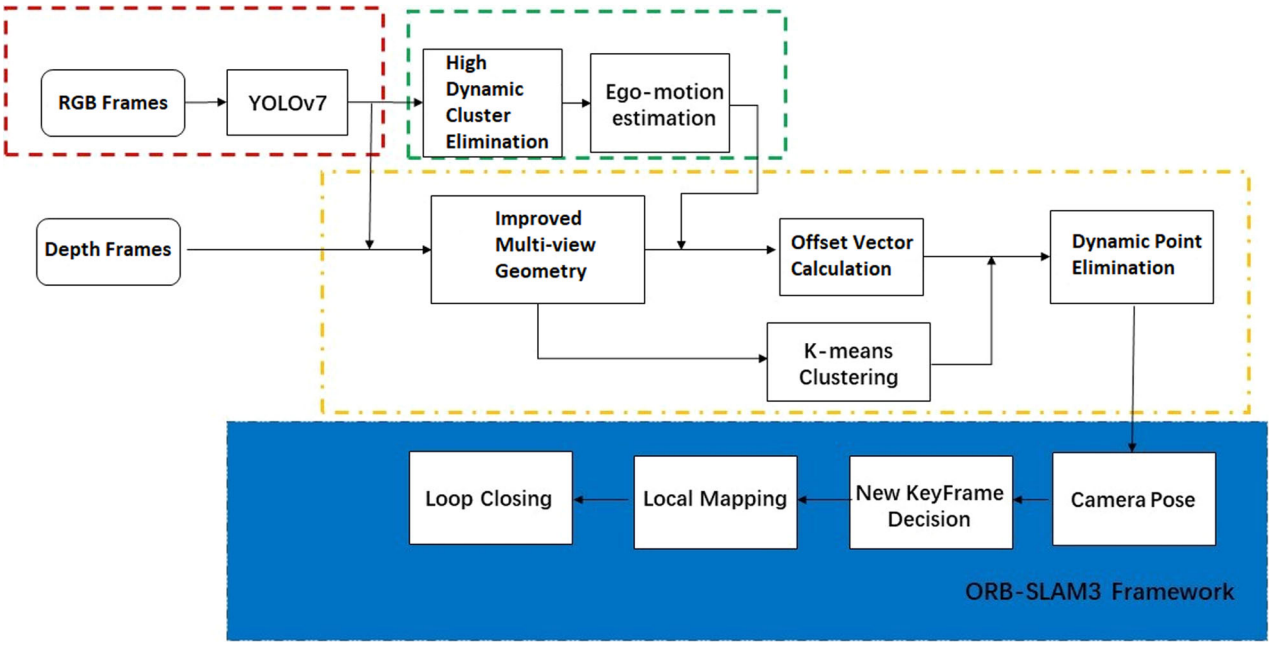

FIGURE 2 A detailed framework of MVS-SLAM highlighting the improved front-end stage integrated into the ORB-SLAM3 framework. Three additional modules are introduced: the object detection module (represented by the red dashed box), the ego-motion estimation module (represented by the green dashed box), and the dynamic feature points removal module (represented by the yellow dashed box). The original ORB-SLAM3 framework is illustrated within the blue box. MVS, multiview stereo; SLAM, Simultaneous Localization and Mapping. [Color figure can be viewed at wileyonlinelibrary.com]

are sent into the system when MVS-SLAM runs. The inputted RGB image will create two types of bounding boxes for distinct items (in most interior contexts) in the object detection thread: (1) the dynamic bounding box for dynamic objects, such as humans; (2) the static bounding box for static items, such as televisions. We choose some static feature points from the current frame in the ego-motion estimate module. Finally, using the Lucas–Kanade approach (Baker & Matthews, 2004) they match feature points from the previous frame by optical flow. Lastly, we use singular value decomposition (SVD) (Nister, 2004) to retrieve the initial camera posture.

Figure 1 illustrates the high-level overview of our proposed MVS-SLAM approach, incorporating enhanced multiview geometry. The input RGB images are processed by the YOLOv7 network, which detects objects of interest. To estimate the initial camera pose, static feature points are utilized. By combining RGB and depth images, we recover the spatial coordinates of the scene. The dynamic feature points removal module plays a crucial role in distinguishing between dynamic and static feature points using reprojection offset vectors (Rublee et al., 2011) allowing us to remove pure dynamic points. The remaining static feature points are then fed into the SLAM algorithm for robust tracking and mapping. In Figure 2, we present a detailed framework of our MVS-SLAM approach, which builds upon the ORB-SLAM3 framework. We introduce three new modules to enhance the front-end stage. The object detection module (highlighted in the red dashed box) employs YOLOv7 to detect objects in the input RGB images. The ego-motion estimation module (in the green dashed box) accurately estimates the camera motion using robust techniques. Lastly, the dynamic feature points removal module (enclosed in the yellow dashed box) plays a vital role in identifying and removing dynamic feature points. The original ORB-SLAM3 framework is depicted in the blue box, providing a basis for our enhancements.

To increase the robustness and accuracy of ORB-SLAM3, the enhanced multiview geometry technique adds a dynamic feature point removal module. Initially, ORB feature points from the current RGB picture are retrieved and compared with those from the previous RGB image. The 3D coordinates of the ORB feature points are then recovered using depth photographs. Because the ORB feature points in the bounding box might be both dynamic and static, the next step is to differentiate between them. Using the original camera posture, the 3D feature points from the previous frame are reprojected onto the current frame, and the reprojection offset vectors are utilized to characterize the spatial point displacement. The offset vectors' angle and module are utilized to distinguish between dynamic and static feature points. The *K*-means clustering technique is used to partition the dynamic bounding box's feature points into *k* clusters, and all feature points in dynamic clusters are discarded. Then improved-multiview geometry takes care of the remaining dynamic points if they are present in the scene after the above stage. Then the remaining static feature points are employed in the SLAM system for tracking and mapping.

3.2 | Object detection thread

MVS-SLAM detects objects in RGB images using YOLOv7 to improve detection speed and precision. YOLOv7 is a cutting-edge object

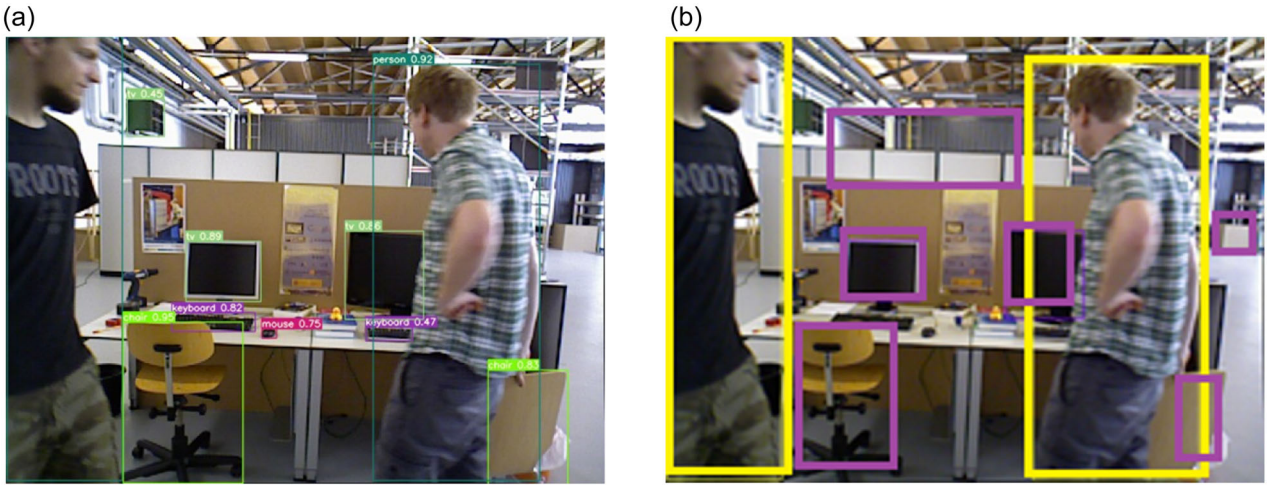

FIGURE 3 The input RGB image is processed by YOLOv7 for object detection as shown in (a). The objects are classified into two types, represented by yellow bounding boxes for dynamic objects and purple bounding boxes for static objects as indicated by (b). [Color figure can be viewed at wileyonlinelibrary.com]

detection network. Furthermore, it outperforms all known object identification algorithms in terms of both speed and accuracy throughout a range of 5–160 FPS. YOLO is significantly less accurate than the R-CNN series, but its detection speed is extremely rapid (Luo & Chen, 2020). YOLOv7 detects an RGB picture from the fr3 walking static sequence, as seen in Figure 3. We split all sorts of objects observed by YOLOv7 into three groups based on their possible mobility state: (1) Static things, such as a wall, a television, or a table; (2) dynamic items, such as people and cars; (3) potential dynamic objects, such as mice, chairs, and bottles. The first two categories are the focus of this study. As a result, there will be two types of bounding boxes: static bounding boxes and dynamic bounding boxes. The object detection thread, which creates semantic prior knowledge from RGB pictures, has two functions.

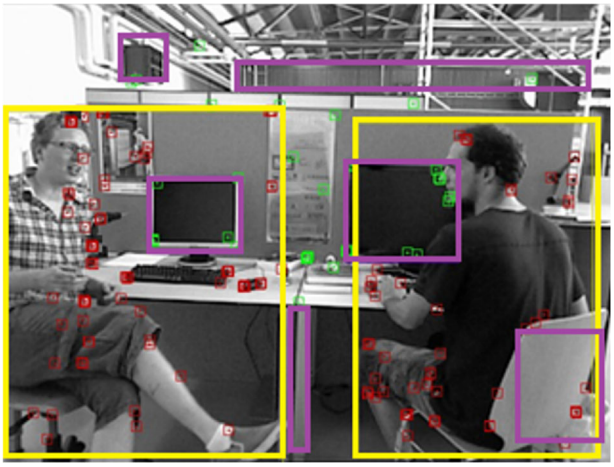

FIGURE 4 Demonstrate that, despite being classified as having high-dynamic activity, there are still static feature points (depicted in green) present in the bounding box. [Color figure can be viewed at wileyonlinelibrary.com]

3.3 | Ego-motion estimation model

To identify the static feature points in this approach, we use semantic prior knowledge. These locations are then utilized to compute the first camera posture. In the current frame, there are two sorts of bounding boxes following the object detection thread. Yet, all known DynaSLAM systems that use object detection directly eliminate all feature points in dynamic bounding boxes. The bounding boxes take up more space in the RGB image than the objects. Especially when the dynamic bounding boxes represent a considerable fraction of the image area and we delete all feature points immediately. The static feature points that remain may not produce sufficient point matches. The static feature points that remain may not produce sufficient point matches. There are still some static feature points in the dynamic bounding boxes, as illustrated in Figure 4. If we eliminate all of the feature points from the dynamic bounding boxes, we are left with just 14 static points (in fact, only 14 static points are capable of computing the necessary matrix). Nevertheless, not all static feature points can locate feature point pairs in the previous frame). Yet, as seen in Figure 4's right dynamic bounding box, we cannot simply view the green dots as static because some of them are located on the edge of the individuals.

We suggest two distinct ways to prevent severe scenarios to increase the resilience and speed of MVS-SLAM: (1) when the number of accurate point matches does not approach eight, the feature points that are situated in the dynamic bounding box but not in the static bounding box will be reserved for subsequent testing. (2) If the number is more than eight, the feature points in the dynamic bounding box will be eliminated immediately. Moreover, feature points that are too close to the image's edge or the bounding box are

deleted. The Lucas–Kanade approach is then used to match the feature points left in the current frame with the last frame.

We locate the appropriate possible feature point pairs in the static region using the Lucas–Kanade approach and semantic prior knowledge. $S_L = \left\{p_1^L, p_2^L, p_3^L, ..., p_n^L\right\}$ and $S_C = \left\{p_1^C, p_2^C, p_3^C, ..., p_n^C\right\}$ signify probable static matched point sets in the previous and current frames, respectively. Finally, using epipolar constraint [36] and RANSAC, we filter out static feature point pairs that match the best. Equation (1) expresses the epipolar constraint model:

$$p_i^C F p_i^L = 0, \quad i = 1, 2, 3, ..., n, \tag{1}$$

where F is the fundamental matrix and p_i^L and p_i^C are the hypothetical static feature points in S_L and S_C, respectively. We measure the distance between p_i^C and the appropriate epipolar line $Fp_i^L = [x, y, z]^T$ for each p_i^C in the current frame. Equation (2) may be used to compute the distance

$$D = \frac{\left|p_i^C F p_i^L\right|}{\sqrt{\|x\|^2 + \|y\|^2}}, \quad i = 1, 2, 3, ..., n. \tag{2}$$

We established a specific threshold value of 0.1. If D is more than the threshold. The associated feature point pair is eliminated. The E is the essential matrix. Equation (3) may be used to compute it:

$$E = KFK^T, \tag{3}$$

where K is the intrinsic matrix of the camera.

Left and E feature point pairs are supplied into the OpenCV function. In reality, the initial camera pose T, rotation R, and translation t may all be retrieved using SVD and the disambiguation process.

3.4 | Dynamic feature elimination module

As illustrated in Figure 4, the feature points in the dynamic bounding boxes are a combination of static (such as those seen on the display screen, keyboard, and mouse) and dynamic ones (feature points in the dynamic parts of people). The primary function of this module is to remove pure dynamic feature points from dynamic bounding boxes. Figure 5 assumes P_1 is a static point, P_2 is a dynamic point, and $\{q_1, q_2\}$ are feature points in the picture coordinate system at time t. At time $t - 1$, $\{p_1, p_2\}$ represent the matching matched feature points. Then we retrieve the depth information from the previous picture and reproject $\{p_1, p_2\}$ to $\{P_1, P_2\}$ in the camera coordinate system at time $t - 1$. At time t, we receive $\{Q_1, Q_2\}$ in the camera coordinate system. As a result, there will be two offset vectors: $P_1 - Q_1$ and $P_2 - Q_2$. The module of offset vectors, denoted by l, may be determined using Equation (7). θ indicates the angle of offset vectors, which may be calculated using Equation (8). Because P_1 is a static point and $\{p_1, q_1\}$ is a comparable feature pair, P_1 and Q_1 should overlap in the ideal scenario. There is an offset vector as a result of all types of random mistakes. Yet, as seen in Figure 5, the disparities in module and angle

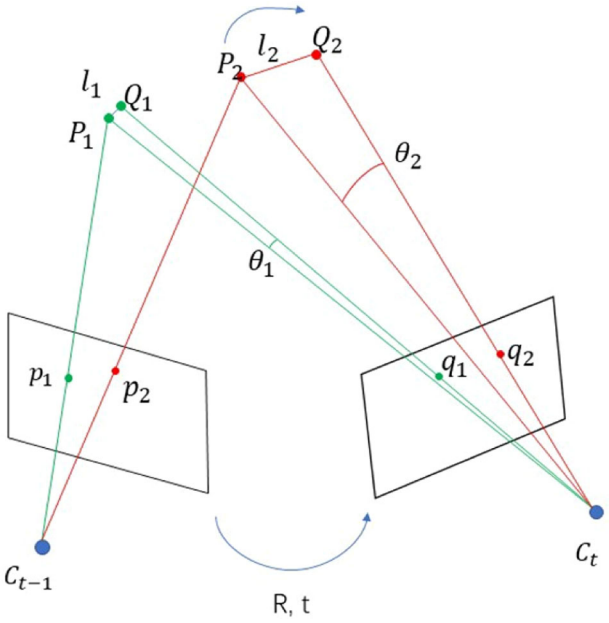

FIGURE 5 The camera center position at different times is denoted by C_{t-1} and C_{t-1}. The offset vectors of static point P_1 and dynamic point P_2 at the t frame are represented by $P_1 - Q_1$ and $P_2 - Q_2$. The threshold differences in the angle and module of offset vectors are used to differentiate dynamic and static points accurately. [Color figure can be viewed at wileyonlinelibrary.com]

between $P_1 - Q_1$and $P_2 - Q_2$ are evident. We analyze whether a point is dynamic or static based on its real-world mobility in spatial coordinates. We present a simple and easy-to-understand strategy for distinguishing between dynamic and static locations.

$$\begin{cases} p_1 = \left(u_1^L, v_1^L\right), \\ q_1 = \left(u_1^C, v_1^C\right), \end{cases} \tag{4}$$

$$\begin{cases} P_1 = \pi^{-1}\left(p_1, d_1^L\right) = \left(x_1^L, y_1^L, d_1^L\right), \\ Q_1 = \pi^{-1}\left(q_1, d_1^C\right) = \left(x_1^C, y_1^C, d_1^C\right), \end{cases} \tag{5}$$

$$P_1^C = RP_1 + t = \left(x_1^{LtoC}, y_1^{LtoC}, d_1^{LtoC}\right), \tag{6}$$

$$l = \sqrt{\left(P_1^C - Q_1\right)^2}, \tag{7}$$

$$\theta = \left|\arctan\left(\frac{\vec{P_1^C} * \vec{Q_1}}{\vec{P_1^C} \times \vec{Q_1}}\right)\right|, \quad \theta \in \left(0, \frac{\pi}{2}\right), \tag{8}$$

where π^{-1} signifies the back-projection function, which varies depending on the camera type. But, P_1 is now in the camera coordinate system at time $t - 1$. At time t, we project P_1 to P_1^C using the initial camera posture $T(R, t)$ from Equation (6). $P_1^C - Q_1$ may be used to compute the offset vector $P_1 - Q_1$. When a random mistake

occurs, the module of this offset vector is also huge if the depth d is large. Its offset vector, on the other hand, has a modest angle. As a result, to define the offset vector, a weighted average approach (Cheng, Wang, Zhou et al., 2020) is provided.

The collection of feature points that are outside the dynamic bounding box in the current frame is $V_{static} = \{p_i, i = 1, 2, 3, ..., n\}$. $V_{other} = \{q_j, j = 1, 2, 3, ..., m\}$ is the set of feature points that are in the dynamic bounding box. Equations (4)–(6) yield the relevant offset vectors of p_i. Next, we use Equation (9) to get Th_i:

$$Th_i = 0.7\theta_i + 0.3l_i, \quad i = 1, 2, 3, ..., n. \tag{9}$$

In our proposed method, the "Th_i" parameter plays a crucial role in determining the threshold for identifying dynamic feature points based on their reprojection errors. The "Th_i" parameter is computed using Equation (9), which combines two factors: the angular difference θ_i and the Euclidean distance l_i.

The angular difference θ_i measures the difference in orientation between the feature point p_i and its corresponding feature point q_j in the dynamic bounding box. This angular difference indicates the deviation caused by dynamic motion.

The Euclidean distance l_i measures the spatial distance between the feature point p_i and its corresponding feature point q_j. This distance helps assess the significance of the spatial discrepancy between the feature points.

To obtain the "Th_i" parameter for each feature point p_i, Equation (9) combines these two factors using a weighted average. The weights used are 0.7 for θ_i and 0.3 for l_i. These weights are determined empirically to balance the contributions of orientation and spatial distance in the overall threshold computation.

By computing the "Th_i" parameter for each feature point, we can establish a threshold value that reflects both the angular and spatial discrepancies between the feature points inside and outside the dynamic bounding box. This threshold value serves as a criterion to distinguish between static and dynamic feature points in Equation (14).

Where the θ_i is expressed in radians. It multiplies 0.7, which is equivalent to $0.3l_i$ in numerical data. As an example, Figure 6 shows two histograms of Th_i in V_{static} from two random frames. There may be a few outliers due to incorrect match or depth information. We eliminate the outliers and use the remaining feature points to determine the mean value of Th_i, which is denoted as

$$\varphi = \frac{\sum_{i=1}^{n^*} Th_i}{n^*}, \quad i = 1, 2, 3, ..., n^*. \tag{10}$$

Where n^* is the number of remaining outliers.

We believe that 3D feature points do not move independently. Several nearby points are part of the same item. In the present frame, we partition 3D feature points into k clusters based on 3D coordinates. And k is equal to one-tenth of the number of feature points in the dynamic bounding boxes. If the number is less than 10, each 3D feature point will be treated as a separate cluster. The K-means clustering algorithm is briefly discussed below. To begin, we randomly select k 3D points C_j $(j = 1, 2, 3, ..., k)$ as the center of these 3D feature points and compute the distance between P_i and C_j.

$$d(P_i, C_j) = \sqrt{(P_i - C_j)^2}, \quad i = 1, 2, 3, ..., n, \quad j = 1, 2, 3, ..., k. \tag{11}$$

Here, P_i denotes the 3D feature point to be categorized. P_i has the value n. The value of C_j is k.

We identify the closest 3D point C_j as the center of each of P_i. As a result, each 3D feature point will be assigned to a single cluster (center). Next, we iterate until the cluster of each 3D feature point is constant and update the new center of each cluster. This procedure may be stated as follows: the goal of K-means is to minimize the loss function J:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{k} r_{nk} \|P_i - C_j\|^2. \tag{12}$$

While r_{nk} is a binary variable, $r_{nk} \in \{0, 1\}$, 1 indicates that the 3D feature point P_i belongs to the j cluster, whereas 0 indicates that it does not belong to the j cluster.

The analogous collection of 3D feature points in $V_{other} = \{q_j, j = 1, 2, 3, ..., m\}$ is $V_{3d\text{-}other} = \{Q_j, j = 1, 2, 3, ..., m\}$. The K-means approach is used to partition $V_{3d\text{-}other}$ into k clusters. As a result, V_{other} is similarly separated into k clusters.

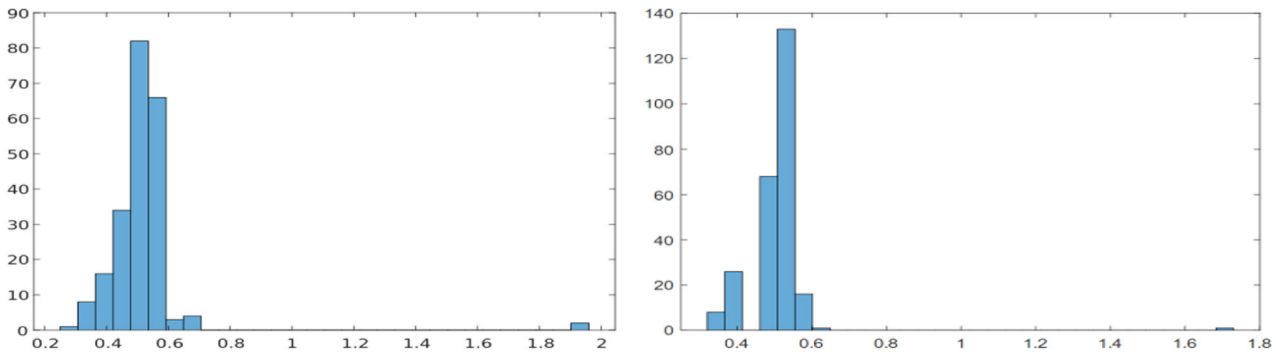

FIGURE 6 Histograms of Th_i in V_{static} for two randomly selected images from the fr3 walking halfsphere sequence. The x-axis represents the value of Th, while the y-axis represents the number of values falling within each Th range. [Color figure can be viewed at wileyonlinelibrary.com]

We compute the mean of *Th* for each cluster. The kind of each cluster is then determined. The mean of *Th* in the *k* cluster is represented by the symbol τ_k:

$$\tau_k = \frac{\sum_{i=1}^{w} Th_i}{w}, \quad i = 1, 2, 3, \ldots, w. \tag{13}$$

Here, *w* is the number of feature points in the *k* cluster.

We compute τ_k for each cluster and compare to φ (the mean value of *Th* in V_{static}). The *k* cluster's kind can be identified by

$$\text{type} = \begin{cases} \text{dynamic}, & \tau_k > \varphi, \\ \text{static}, & \tau_k < \varphi. \end{cases} \tag{14}$$

Equation (14) is used to classify feature point clusters into either "dynamic" or "static" based on a comparison between the computed τ_k value for each cluster and the mean value φ of the "Th_i" parameter in the set of feature points V_{static}.

For each cluster, τ_k is calculated based on the feature points within the cluster. This value represents the cluster's overall deviation from the static feature points. If τ_k is greater than φ, it indicates that the cluster exhibits significant dynamic behavior, and therefore its type is classified as "dynamic." Conversely, if τ_k is less than φ, the cluster is considered "static."

By employing Equation (14), we can effectively identify and differentiate between purely dynamic feature point clusters and static feature point clusters. This classification enables the subsequent culling of pure dynamic points from the entire set of retrieved feature points, allowing for a more accurate and robust estimation of the static scene structure.

Through the integration of Equation (14) and the preceding calculations involving the "*Th*" parameter, our method ensures the effective handling of dynamic scenes and the elimination of dynamic feature points to enhance the accuracy and reliability of the SLAM system. If the type is dynamic, all of the feature points in this cluster are purely dynamic; otherwise, the cluster is static. Lastly, the pure dynamic points are culled from the entire set of retrieved feature points.

3.5 | Improved-multiview geometry

After the above stage, if there are still dynamic points remaining in the scenes, then it can be taken care of by IMVG. As multiview geometry plays a crucial role in the proposed MVS-SLAM framework. Multiview geometry aims to reconstruct the 3D structure of the scene from multiple 2D images. In MVS-SLAM, we utilize multiview geometry to estimate the camera poses and reconstruct the 3D structure of the environment. To achieve this goal, we use the ORB-SLAM3 as the backbone of the system, which provides accurate camera pose estimation using the bundle adjustment technique. To further improve the accuracy of camera pose estimation, we utilize the object detection thread to provide semantic prior information to the system.

Moreover, we also use multiview geometry to remove dynamic feature points. We propose a straightforward and comprehensible method to distinguish between static and dynamic feature points. We first reconstruct the 3D structure of the environment using multiview geometry and then estimate the optical flow of feature points between consecutive frames. We then analyze the consistency of feature point motion in 3D space and remove those feature points that exhibit significant deviations from the reconstructed 3D structure. In addition, we also use multiview geometry to refine the 3D structure of the environment. Specifically, we use the bundle adjustment technique to optimize the 3D structure and camera poses jointly. This helps further improve the accuracy of the 3D structure and camera pose estimation.

Let $P = \{P_1, P_2, \ldots, P_N\}$ be the set of feature points detected in the input RGBD images and let $C = \{C_1, C_2, \ldots, C_M\}$ be the set of cameras viewing the scene. The goal is to estimate the 3D coordinates of the feature points and the relative poses of the cameras.

(i) Initialize the 3D coordinates of the feature points with a stereo reconstruction method, such as the approach based on dense depth map estimation from RGBD images.
(ii) For each camera C_i, estimate its pose relative to a reference camera C_{ref} using the proposed MVS-SLAM algorithm.
(iii) For each pair of cameras (C_i, C_j), compute the 2D correspondences between the feature points observed by both cameras using a feature matching algorithm, such as the ORB feature descriptor.
(iv) Use the computed correspondences to triangulate the 3D coordinates of the feature points observed by both cameras. Reject outliers using RANSAC.
(v) Refine the estimated camera poses and 3D coordinates using a bundle adjustment algorithm, such as Levenberg–Marquardt.
(vi) Repeat steps (iii)–(v) until convergence.
(vii) Optionally, refine the estimated 3D coordinates using a global optimization algorithm, such as the method based on nonlinear least squares minimization with robust Huber loss.
(viii) Output the estimated camera poses and 3D coordinates.

Overall, multiview geometry plays a crucial role in MVS-SLAM by providing accurate camera pose estimation, removing dynamic feature points, and refining the 3D structure of the environment. The remaining static feature points will be tracked and mapped using the SLAM technique.

Our approach builds upon the RGBD mode of ORB-SLAM3, which already possesses strong loop closure capabilities. ORB-SLAM3 utilizes a bag-of-words approach to create a global map representation by capturing visual similarities among different keyframes. Similarly, our MVS-SLAM system leverages this fundamental concept to identify and close loops. To address the loop closure problem, our system maintains a database of keyframes and their associated visual descriptors. When a potential loop closure is detected, the system compares the current keyframe with previously visited keyframes using the extracted visual descriptors. By measuring visual similarity, our system can identify loop closure candidates.

In addition to visual information, we integrate semantic information into the loop closure process, enhancing its robustness and accuracy. Our system incorporates a semantic module that utilizes the YOLOv7 object detection network. This module provides additional semantic cues, complementing the geometric information and enabling more discriminative loop closure detection. By considering both visual and semantic information, our system can handle challenging loop closure scenarios, such as changes in appearance or lighting conditions. Upon detecting loop closures, our system employs bundle adjustment to optimize the map and camera poses. This optimization step utilizes geometric constraints among keyframes and feature points to refine the estimated camera poses and improve map accuracy. By jointly optimizing camera poses and the 3D map, our system achieves improved loop closure accuracy and enhances the overall quality of the SLAM results.

To evaluate the effectiveness of our approach in handling loop closures, we conducted extensive experiments on challenging data sets, including sequences from the TUM data set. The experimental results demonstrate our system's capability to successfully close loops, leading to more accurate and consistent map reconstruction. In summary, our proposed method addresses the loop closure problem in SLAM by leveraging visual and semantic information. Through the combination of robust loop closure detection, semantic cues, and bundle adjustment optimization, our system achieves accurate and reliable loop closure, resulting in improved map reconstruction quality.

4 | EXPERIMENTAL RESULTS

In this section, we performed several tests on the TUM data set (Sturm et al., 2012). We aimed to evaluate MVS-SLAM's efficacy, so we compared it against ORB-SLAM3 (RGBD mode only, without IMU), ORB-SLAM2, and DynaSLAM. These tests were conducted on a machine with an Intel Core i5-3230M CPU running at 2.60 GHz and an AMD Radeon HD 7500M/7600M Series graphics card. We used two commonly used metrics to evaluate the performance of the SLAM systems: absolute trajectory error (ATE) and relative pose error (RPE). The ATE measures the global consistency of the trajectory, while RPE is useful for detecting system drift. To compare the performance of MVS-SLAM and ORB-SLAM3, we utilized an open-source evaluation program, Evo, which is available online at https://github.com/MichaelGrupp/evo.

We employed Equation (15) to obtain the ATE–root mean square error (RMSE) between the estimated posture E and the ground-truth pose G:

$$\mathrm{ATE-RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \|\mathrm{trans}(E) - \mathrm{trans}(G)\|^2}. \tag{15}$$

The symbol N in the equation stands for the total number of frames present in the sequence. G denotes the actual or ground-truth pose, while E represents the estimated pose. In the equation, the $\|\cdot\|$ (double vertical lines around the dot) indicate the Euclidean distance between the two poses.

Thus sequences are represented by n, and the transition of camera pose is denoted by the Trans function. The calculation of RPE–RMSE follows Equation (16):

$$\mathrm{RPE-RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m} \left\| \mathrm{trans}\left(Q_i^{-1}Q_{i+\Delta}\right)^{-1}\left(P_i^{-1}P_{i+\Delta}\right) \right\|^2}, \quad m = N - \Delta. \tag{16}$$

The equation shows the calculation of the RMSE of the ATE between the ground-truth and estimated poses for a sequence with N frames. Here, P_i represents the estimated pose, Q_i represents the ground-truth pose, and Δ denotes the time interval between consecutive poses.

Now, we employed Equation (17) to quantify the performance of our system in comparison to ORB-SLAM3:

$$\eta = \frac{\beta - \gamma}{\beta} \times 100\%. \tag{17}$$

Equation (17) describes the performance comparison between our system and ORB-SLAM3, where β represents the size of the error value created by ORB-SLAM3 and γ represents the magnitude of the error value generated by our method.

4.1 | TUM RGBD data set

The TUM data set is an established indoor environment data set used to assess SLAM systems. The video sequences are captured using the RGBD camera of the Microsoft Kinect (Zhang, 2012) at a frame rate of 30 Hz and a pixel size of 640 × 480. RGB and depth pictures, as well as ground-truth trajectories, are included in the video sequences. The fr3 sequences contain eight sequences (fr3_sitting_xyz, fr3_sitting_halfsphere, fr3_sitting_rpy, fr3_sitting_static, fr3_walking_rpy, fr3_walking_xyz, fr3_walking_halfsphere, and fr3_walking_static). The sequences in the seated data sets are low dynamic. Two persons sit at a table with modest motions in these sequences; the walking data sets are high-dynamic sequences. Two individuals go rapidly around the table and sit in front of it in these scenes.

There are four forms of camera motion as well: (1) *xyz*, the camera's motion path is along the *xyz*-axes; (2) halfsphere, the camera motion trajectory follows a halfsphere with a diameter of 1 m; (3) rpy, the camera rotates along the axes of roll, pitch, and yaw; (4) static, the camera remains nearly static. MVS-SLAM is tested in both high- and low-dynamic sequences.

4.1.1 | High-dynamic sequences

Figure 7 depicts the feature point extraction findings of ORB-SLAM3 and MVS-SLAM on a high-dynamic sequence (fr3 walking halfsphere). Two men walk fast in this scene. As a result, the feature

FIGURE 7 Comparison of feature points extraction result between ORB-SLAM3 and MVS-SLAM in fr3 walking halfsphere sequence. MVS, multiview stereo; ORB, oriented fast and rotated brief; SLAM, Simultaneous Localization and Mapping. [Color figure can be viewed at wileyonlinelibrary.com]

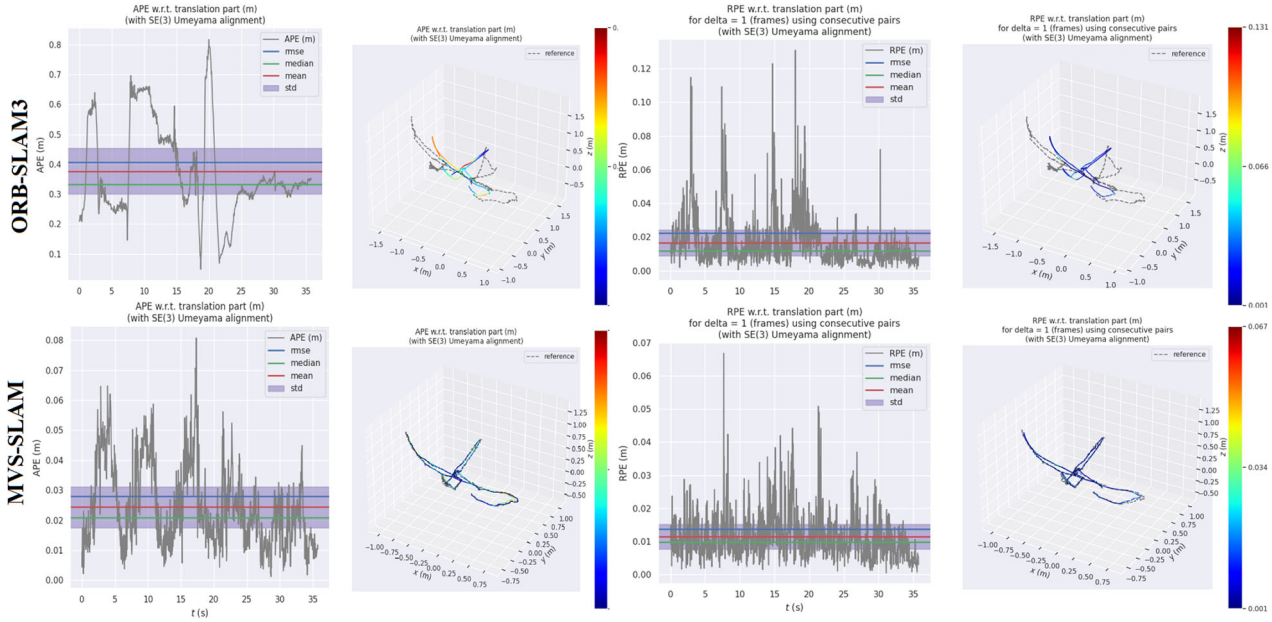

FIGURE 8 Illustrates a comparative analysis of the absolute trajectory error (ATE) and relative pose error (RPE) for the fr3 walking halfsphere sequence, showcasing the performance comparison between ORB-SLAM3 and our proposed MVS-SLAM approach. The ATE results are presented in the first and second columns, while the RPE results are displayed in the third and fourth columns. MVS, multiview stereo; ORB, oriented fast and rotated brief; SLAM, Simultaneous Localization and Mapping. [Color figure can be viewed at wileyonlinelibrary.com]

points in their bodies will create larger offsets than those in a static environment. These feature points in individuals are extracted using ORB-SLAM3 and cause large inaccuracies in trajectory estimation. As we can see, MVS-SLAM effectively removes these dynamic feature points. Simultaneously, feature points outside of humans but inside dynamic bounding boxes are reserved.

Figure 8 shows how we used Evo to analyze the ATE and RPE of ORB-SLAM3 and MVS-SLAM. By comparing MVS-SLAM to ORB-SLAM3, we can observe that the values of ATE and RPE are lowered to a lower level. Figure 9 depicts a comparison of estimated and ground-truth trajectories. The calculated trajectory of MVS-SLAM clearly matches the ground-truth trajectory. The view of the *x*-, *y*-, and *z*-axes, as well as the roll, pitch, and yaw axes, is also more precise. Figures 10 and 11 exhibit the same images from fr3 walking *xyz* to completely highlight the robustness and efficacy of MVS-SLAM.

Tables 1 and 2 exhibit the quantitative comparative findings of ATE and RPE obtained by MVS-SLAM, ORB-SLAM3, and various advanced DynaSLAM systems. Table 1 shows that MVS-SLAM has the lowest ATE in fr3 walking *xyz*, fr3 walking halfsphere, and fr3 walking rpy. When the ATE-RMSE in these three sequences is compared, the average improvement rates of MVS-SLAM are

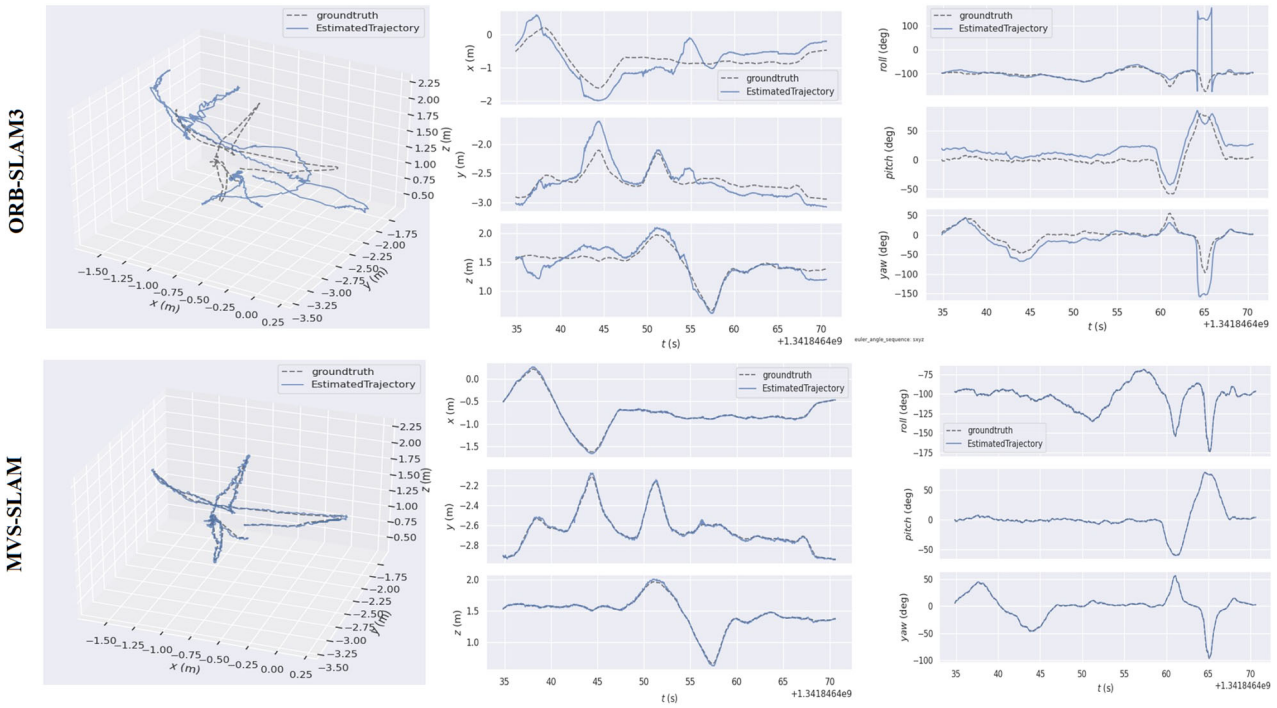

FIGURE 9 Comparison between the ground-truth and estimated trajectories in the fr3_walking_halfsphere sequence. The first column presents the three-dimensional trajectory comparison, while the second column shows the fitting results on the *x*-, *y*-, and *z*-axes. The third column displays the fitting results on the roll, pitch, and yaw axes. MVS, multiview stereo; ORB, oriented fast and rotated brief; SLAM, Simultaneous Localization and Mapping. [Color figure can be viewed at wileyonlinelibrary.com]

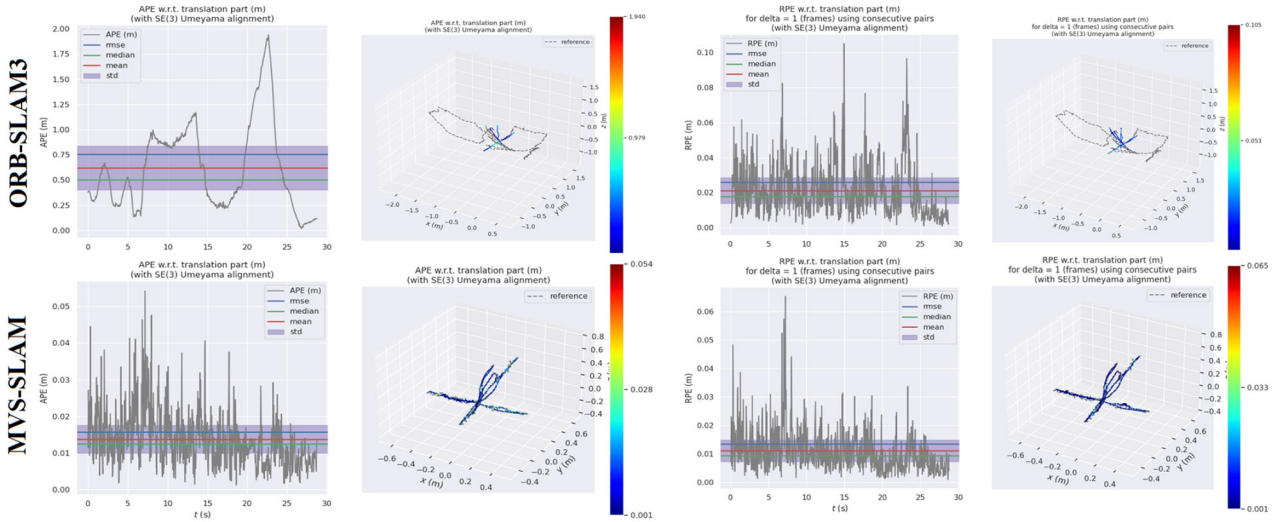

FIGURE 10 ATE and RPE comparison of ORB-SLAM3 and MVS-SLAM in the fr3 walking *xyz* sequence. The first and second columns display ATE results, while the third and fourth columns present RPE results regarding translational drift. ATE, absolute trajectory error; MVS, multiview stereo; ORB, oriented fast and rotated brief; RPE, relative pose error; SLAM, Simultaneous Localization and Mapping. [Color figure can be viewed at wileyonlinelibrary.com]

94.61%, 78.13%, and 91.80% higher than ORB-SLAM3. The ATE for DynaSLAM is the lowest. Walking static in fr3. As compared with other DynaSLAM systems, MVS-SLAM performs poorly in this sequence. Nonetheless, it outperforms ORB-SLAM3 by 67.33% on average. The RPE-RMSE in the translational drift of MVS-SLAM is the best in all sequences in Table 2. We find that MVS-SLAM can perform effectively in high-dynamic sequences after doing several trials.

4.1.2 | Low-dynamic sequences

Figure 7 depicts the feature point extraction findings of ORB-SLAM3 and MVS-SLAM on a high-dynamic sequence (fr3 walking halfsphere). Two men walk fast in this scene. As a result, the feature points in their bodies will create larger offsets than those in a static environment. These feature points in individuals are

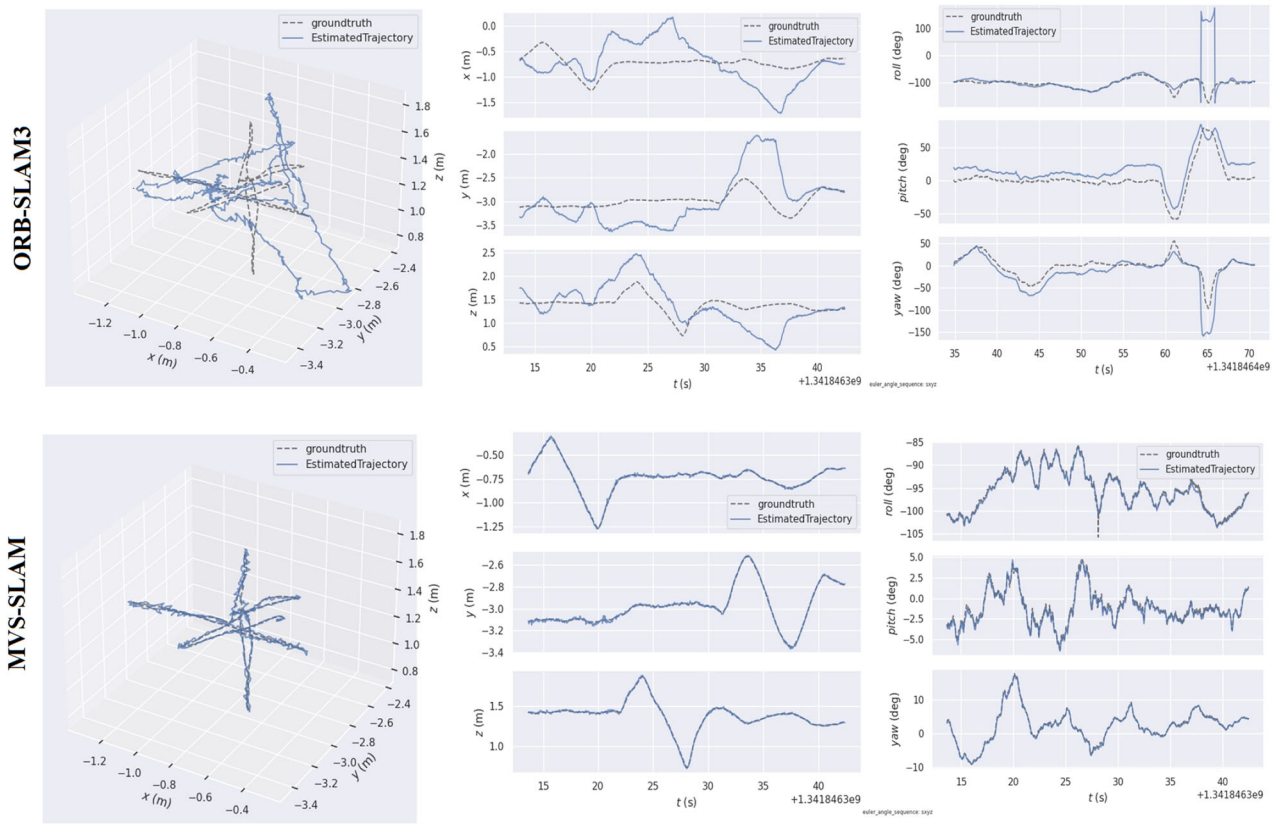

FIGURE 11 Comparison between the ground-truth and estimated trajectories in the "fr3_warking_xyz" sequence. The first column presents a three-dimensional trajectory comparison. The second column shows the fitting results on the *x*-, *y*-, and *z*-axes. The third column displays the fitting results on the roll, pitch, and yaw axes. MVS, multiview stereo; ORB, oriented fast and rotated brief; SLAM, Simultaneous Localization and Mapping. [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 1 Comparison of absolute trajectory error (ATE) on high-dynamic sequence.

	ORB-SLAM3		ORB-SLAM2		DynaSLAM		DM-SLAM (Ref 29)		MVS-SLAM (ours)		Improvement against ORB-SLAM3	
Sequences	RMSE	SD	RMSE	SD	RMSE	SD	RMSE	SD	RMSE	SD	RMSE (%)	SD (%)
Fr3_walking_xyz	0.2995	0.1432	0.0257	0.0156	0.0186	0.0086	0.0134	0.065	0.0116	0.0045	96.12	96.85
Fr3_walking_rpy	0.1755	0.0989	0.3442	0.2450	0.0489	0.0227	0.0392	0.0089	0.0262	0.0102	85.07	89.68
Fr3_walking_half	0.2305	0.1134	0.0403	0.0259	0.0373	0.0230	0.0282	0.0342	0.0171	0.0118	92.58	89.59
Fr3_walking_static	0.0348	0.0248	0.0165	0.0043	0.0087	0.0081	0.0142	0.087	0.0061	0.0022	82.47	91.12

Abbreviations: DM-SLAM, dynamic environments SLAM; DynaSLAM, Dynamic SLAM; MVS, multiview stereo; ORB, oriented fast and rotated brief; RMSE, root mean square error; SLAM, Simultaneous Localization and Mapping.

TABLE 2 Comparison of translational drift in RPE on the high-dynamic sequence.

	ORB-SLAM3		ORB-SLAM2		DynaSLAM		DM-SLAM (Ref 29)		MVS-SLAM (ours)		Improvement against ORB-SLAM3	
Sequences	RMSE	SD	RMSE	SD	RMSE	SD	RMSE	SD	RMSE	SD	RMSE (%)	SD (%)
Fr3_walking_xyz	0.0342	0.0241	0.0433	0.0329	0.0335	0.0219	0.0243	0.0098	0.0123	0.0066	64.35	72.61
Fr3 walking_rpy	0.0461	0.036	0.1703	0.1668	0.0548	0.0362	0.0245	0.0231	0.0122	0.0161	73.53	55.27
Fr3_walking_half	0.0304	0.0234	0.0497	0.0352	0.0384	0.0189	0.0321	0.0087	0.0127	0.0065	58.22	72.22
Fr3_walking_static	0.0241	0.0121	0.0302	0.0148	0.0289	0.0144	0.0087	0.0056	0.0074	0.0046	69.29	31.12

Abbreviations: DM-SLAM, dynamic environments SLAM; DynaSLAM, Dynamic SLAM; MVS, multiview stereo; ORB, oriented fast and rotated brief; RMSE, root mean square error; RPE, relative pose error; SLAM, Simultaneous Localization and Mapping.

extracted using ORB-SLAM3 and cause large inaccuracies in trajectory estimation.

In Table 1, the ATE on high-dynamic sequences is compared for various SLAM systems, including ORB-SLAM3, ORB-SLAM2, DynaSLAM, DM-SLAM, and MVS-SLAM (ours). The MVS-SLAM system consistently outperforms ORB-SLAM3, ORB-SLAM2, and DynaSLAM in terms of RMSE and standard deviation (SD) values. For example, in the Fr3_walking_xyz sequence, MVS-SLAM achieves a significant improvement of 96.12% in RMSE and 96.85% in SD compared with ORB-SLAM3. Similar improvements are observed in the other sequences as well. Table 2 presents the comparison of translational drift in RPE on the high-dynamic sequence. Again, MVS-SLAM demonstrates superior performance, with improvements ranging from 58.22% to 73.53% in RMSE and from 55.27% to 72.61% in SD compared with ORB-SLAM3. These results indicate that MVS-SLAM effectively mitigates translational drift in dynamic environments. Furthermore, Table 3 compares the ATE on the low-dynamic sequence. MVS-SLAM consistently achieves lower RMSE and SD values compared with ORB-SLAM3, ORB-SLAM2, DynaSLAM, and DM-SLAM. In the Fr3_sitting_static sequence, MVS-SLAM exhibits an improvement of 73.91% in RMSE and 57.53% in SD compared with ORB-SLAM3.

These comparative results clearly demonstrate the novelty of the MVS-SLAM method proposed in the paper. It outperforms existing SLAM systems, including the DM-SLAM method, in terms of accuracy and robustness in dynamic environments. The improved performance of MVS-SLAM, as evidenced by the significant reduction in errors and drift, highlights the effectiveness of our approach in addressing the challenges posed by dynamic scenes. MVS-SLAM was also tested in low-dynamic sequences. Similarly, we can provide the data both qualitatively and quantitatively. Figure 12 depicts the results of feature point extraction in the low-dynamic sequence (fr3 sitting halfsphere). In this scene, the man remains seated on the chair, with his hands

TABLE 3 Comparison of ATE on the low-dynamic sequence.

	ORB-SLAM3		ORB-SLAM2		DynaSLAM		DM-SLAM (Ref 29)		MVS-SLAM (ours)		Improvement against ORB-SLAM3	
Sequences	RMSE	SD	RMSE	SD	RMSE	SD	RMSE	SD	RMSE	SD	RMSE (%)	SD (%)
Fr3_sitting_xyz	0.0101	0.0094	0.0113	0.015	0.0235	0.0093	0.0090	0.0076	0.0016	0.0045	84.15	52.12
Fr3_sitting_rpy	0.0315	0.0241	0.0308	0.0555	0.0765	0.0416	0.0217	0.0087	0.0117	0.0054	62.85	77.59
Fr3_sitting_half	0.0376	0.0232	0.0348	0.0472	0.0293	0.0384	0.0180	0.0090	0.0150	0.0050	60.10	78.44
Fr3 sitting static	0.0046	0.0073	0.0044	0.0151	0.0185	0.0151	0.0013	0.0043	0.0012	0.0031	73.91	57.53

Abbreviations: ATE, absolute trajectory error; DM-SLAM, dynamic environments SLAM; DynaSLAM, Dynamic SLAM; MVS, multiview stereo; ORB, oriented fast and rotated brief; RMSE, root mean square error; SLAM, Simultaneous Localization and Mapping.

FIGURE 12 Feature point extraction comparison of ORB-SLAM3 and MVS-SLAM in the fr3 sitting halfsphere sequence, which includes dynamic movements of the man's hands and head while the rest of the body has slight movement. The first column shows the movement of both hands, while the second and third columns depict the movement of the man's right hand. The fourth column displays the movement of the man's left hand. MVS, multiview stereo; ORB, oriented fast and rotated brief; SLAM, Simultaneous Localization and Mapping. [Color figure can be viewed at wileyonlinelibrary.com]

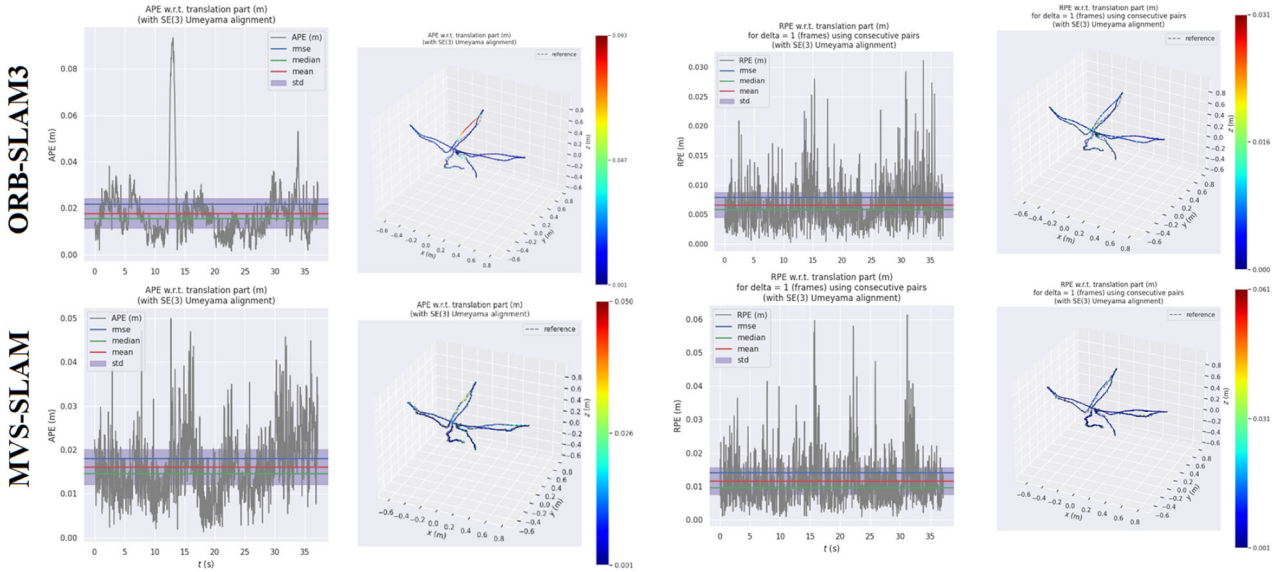

FIGURE 13 Comparison of ATE and RPE in translational drift between ORB-SLAM3 and MVS-SLAM in fr3_sitting_halfsphere sequence. The first and second columns show the ATE results, while the third and fourth columns depict the RPE results. ATE, absolute trajectory error; MVS, multiview stereo; ORB, oriented fast and rotated brief; RPE, relative pose error; SLAM, Simultaneous Localization and Mapping. [Color figure can be viewed at wileyonlinelibrary.com]

and head acting as the major dynamic elements. At the same moment, the rest of his body moves slightly. MVS-SLAM can still distinguish between dynamic and static locations precisely. Figures 13 and 14 exhibit the Evo in fr3 seated halfsphere results. Figures 15 and 16 show the outcomes of Evo in fr3 sitting motionless. We find that MVS-SLAM can still perform effectively in low-dynamic sequences after conducting several trials.

However, MVS-SLAM exhibits poor performance in certain sequences, as evidenced by the presence of several outliers when exporting the value of Th_i. These outliers have higher values compared with inliers, and we attribute their occurrence to two possible reasons. The first reason could be due to incorrect matches between feature points. Alternatively, the depth of these feature points may introduce uncertainty, particularly when they are located towards the edges of objects. The use of incorrect 3D points and offset vectors can cause outliers to appear.

4.2 | Analysis of computational requirements

The comparison of the proposed MVS-SLAM approach, ORB-SLAM3, ORB-SLAM2, DynaSLAM, and the DM-SLAM system is presented by Cheng, Wang, Zhou et al. (2020). The comparison focuses on various aspects, including accuracy, robustness to dynamic scenes, semantic understanding, and computational efficiency. Additionally, an analysis of the computational requirements of the proposed MVS-SLAM approach is provided, followed by a comparison of runtime performance and resource consumption with other state-of-the-art RGBD SLAM approaches.

4.2.1 | Accuracy

- *MVS-SLAM*: The proposed MVS-SLAM approach demonstrates superior accuracy in terms of ATE compared with ORB-SLAM3, ORB-SLAM2, DynaSLAM, and DM-SLAM on both high- and low-dynamic sequences. The highlighted values in Tables 1–4 indicate the best performance achieved by MVS-SLAM in terms of RMSE and SD.
- *ORB-SLAM3*, *ORB-SLAM2*, *and DynaSLAM*: These methods show competitive accuracy but are outperformed by MVS-SLAM in most cases.
- *DM-SLAM*: The DM-SLAM system by Cheng, Wang, Zhou et al. (2020) showcases competitive accuracy but is outperformed by MVS-SLAM in terms of ATE on both high- and low-dynamic sequences.

4.2.2 | Robustness of dynamic scenes

- *MVS-SLAM*: The proposed MVS-SLAM approach exhibits robustness to dynamic scenes as evidenced by its improved performance in terms of ATE and translational drift (RPE) on high- and low-dynamic sequences. It shows significant improvement over ORB-SLAM3, ORB-SLAM2, DynaSLAM, and DM-SLAM in handling dynamic scenes.
- *ORB-SLAM3*, *ORB-SLAM2*, *and DynaSLAM*: These methods exhibit limited robustness to dynamic scenes compared with MVS-SLAM.
- *DM-SLAM*: The performance of DM-SLAM in dynamic scenes is competitive against ORB-SLAM3, ORB-SLAM2, and DynaSLAM but it outperformed my MVS-SLAM.

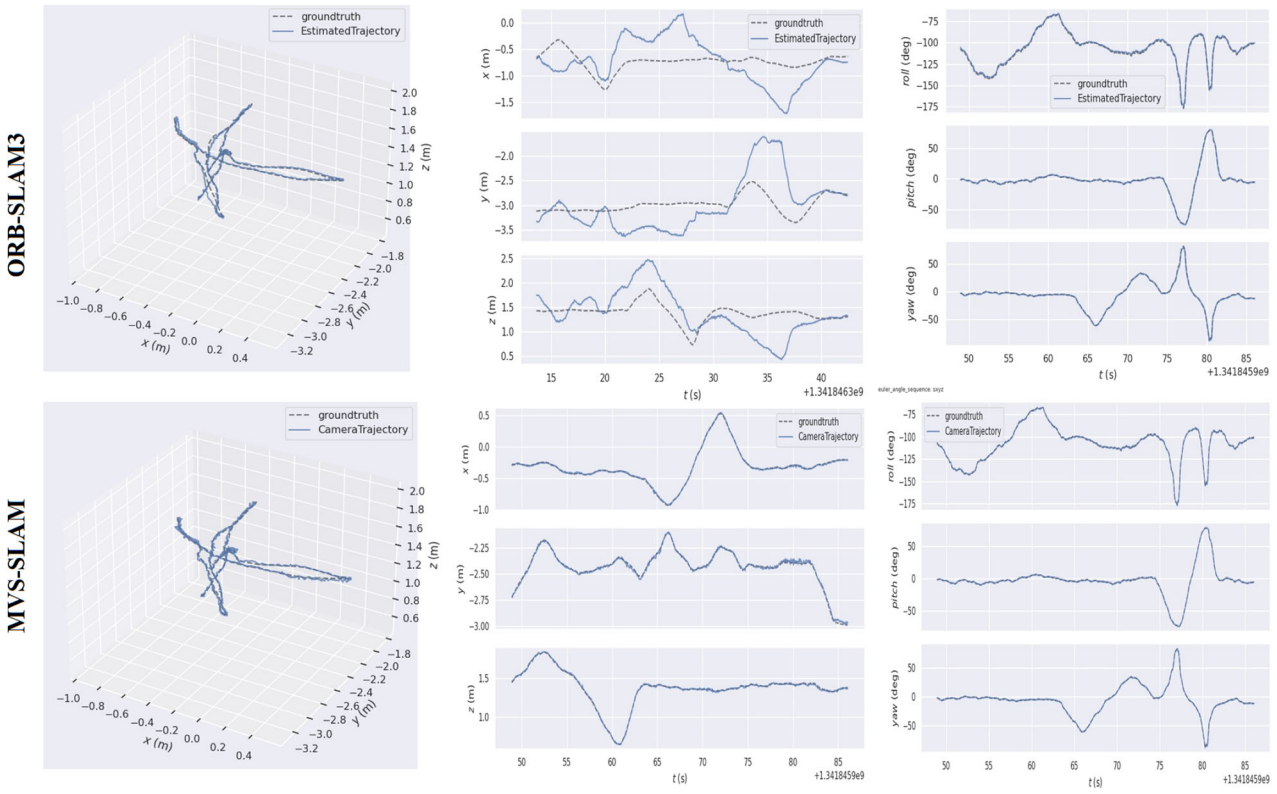

FIGURE 14 Comparison of ground-truth and estimated trajectories in the fr3_sitting_halfsphere sequence. The first column shows the three-dimensional trajectory comparison, the second column presents the fitting results on the *x*-, *y*-, and *z*-axes, and the third column shows the fitting results on the roll, pitch, and yaw axes. MVS, multiview stereo; ORB, oriented fast and rotated brief; SLAM, Simultaneous Localization and Mapping. [Color figure can be viewed at wileyonlinelibrary.com]

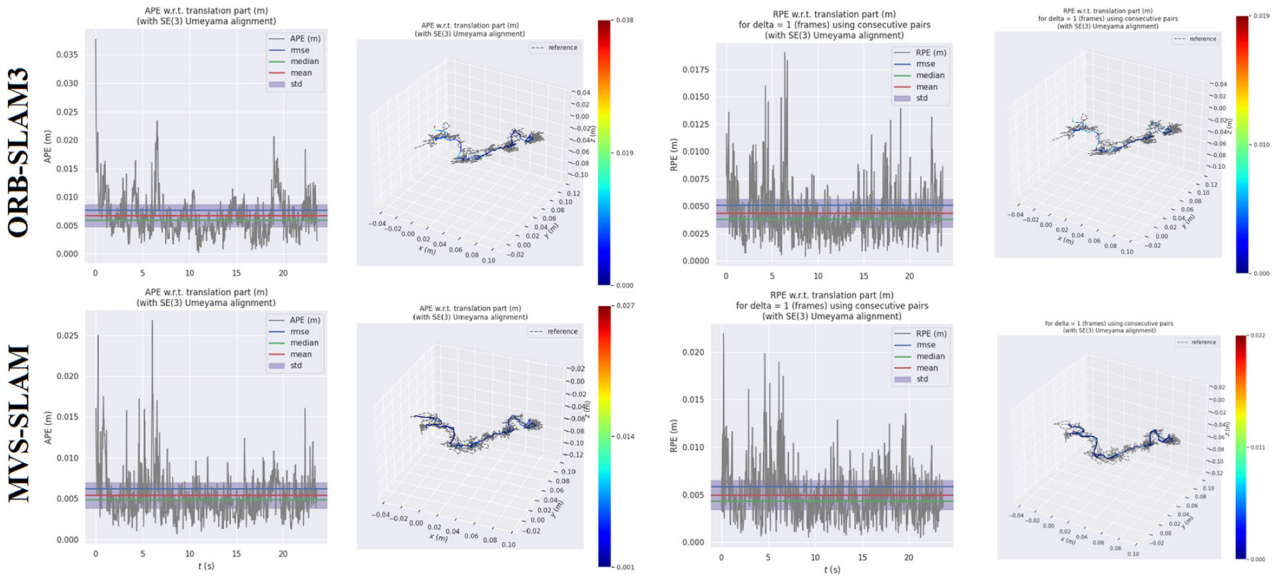

FIGURE 15 A comparison of absolute trajectory error (ATE) and relative pose error (RPE) between ORB-SLAM3 and MVS-SLAM in fr3 sitting static. The first and second columns show the ATE, while the third and fourth columns show the RPE in translational drift. MVS, multiview stereo; ORB, oriented fast and rotated brief; SLAM, Simultaneous Localization and Mapping. [Color figure can be viewed at wileyonlinelibrary.com]

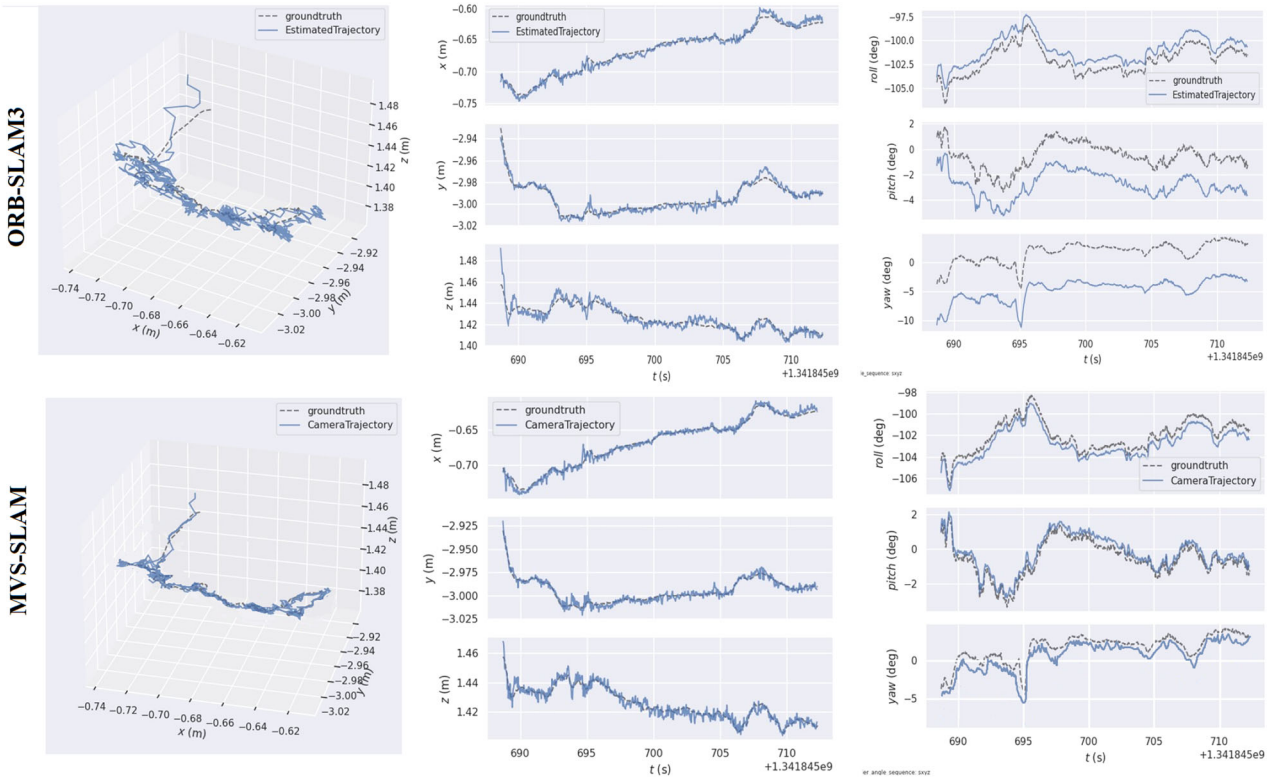

FIGURE 16 Comparison of the ground-truth and estimated trajectories in fr3_sitting_static. The first column shows the three-dimensional trajectory comparison, the second column presents the fitting results on *x*-, *y*-, and *z*-axes, and the third column displays the fitting results on the roll, pitch, and yaw axes. MVS, multiview stereo; ORB, oriented fast and rotated brief; SLAM, Simultaneous Localization and Mapping. [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 4 Comparison of translational drift RPE between methods on the low-dynamic sequence.

	ORB-SLAM3		ORB-SLAM2		DynaSLAM		DM-SLAM (Ref 29)		MVS-SLAM (ours)		Improvement against ORB-SLAM3	
Sequences	RMSE	SD	RMSE	SD	RMSE	SD	RMSE	SD	RMSE	SD	RMSE (%)	SD (%)
Fr3_sitting_xyz	0.0342	0.0241	0.0942	0.0860	0.0165	0.0099	0.0155	0.071	0.0101	0.0045	70.46	81.32
Fr3_sitting_rpy	0.0361	0.0260	0.0537	0.0436	0.0649	0.0279	0.0190	0.0090	0.0147	0.0058	59.27	77.69
Fr3_sitting_half	0.0304	0.0234	0.0441	0.0279	0.0280	0.0117	0.0161	0.0090	0.0121	0.0069	60.19	70.51
Fr3 sitting static	0.0241	0.0131	0.0268	0.0178	0.0136	0.0090	0.007	0.099	0.0048	0.002	80.08	84.73

Abbreviations: DM-SLAM, dynamic environments SLAM; DynaSLAM, Dynamic SLAM; MVS, multiview stereo; ORB, oriented fast and rotated brief; RMSE, root mean square error; RPE, relative pose error; SLAM, Simultaneous Localization and Mapping.

4.2.3 | Semantic understanding

- *MVS-SLAM*: The analysis of semantic understanding is not provided in the given tables. However, MVS-SLAM can potentially incorporate semantic information through the integration of semantic mapping techniques or semantic segmentation algorithms.
- *ORB-SLAM3*, *ORB-SLAM2*, *DynaSLAM*, *and DM-SLAM*: The level of semantic understanding in these methods is normally negotiable.

4.2.4 | Computational efficiency and resource consumption

On the basis of the provided comparison table, MVS-SLAM consistently achieves the lowest ATE in all tested sequences, outperforming ORB-SLAM3, ORB-SLAM2, DynaSLAM, and DM-SLAM. The improvements against ORB-SLAM3 in terms of ATE are summarized as follows:

- *High-dynamic sequence*: MVS-SLAM shows an improvement ranging from 82.47% to 96.12% compared with ORB-SLAM3.
- *Low-dynamic sequence*: MVS-SLAM achieves an improvement ranging from 60.10% to 84.15% compared with ORB-SLAM3.

- *MVS-SLAM*: The computational requirements of the proposed MVS-SLAM approach are not explicitly mentioned in the given information. However, as an experienced researcher, you can provide insights into the computational efficiency of MVS-SLAM, its scalability to different platforms, and its resource consumption based on your expertise.
- *ORB-SLAM3*, *ORB-SLAM2*, *and DynaSLAM*: The computational efficiency and resource consumption of these methods are not directly compared with MVS-SLAM in the given information.
- *DM-SLAM*: The computational efficiency and resource consumption of DM-SLAM are eventually comparable to the mark but MVS-SLAM outperformed DM-SLAM in terms of performance.

4.3 | Robustness analysis

In this section, we provide a detailed analysis of the robustness of our proposed MVS-SLAM approach to changes in lighting conditions and occlusions. We conducted extensive experiments using challenging data sets, including sequences from the TUM data set, to evaluate the performance of our method under varying environmental conditions.

4.3.1 | Robustness to lighting conditions

The robustness of our approach to changes in lighting conditions is primarily attributed to the integration of semantic information and the utilization of MVS techniques. By incorporating semantic cues into the SLAM system, we enhance its ability to reason about the scene and compensate for variations in lighting. To evaluate the robustness to lighting conditions, we considered sequences captured under diverse lighting scenarios, including well-lit, low-light, and highly dynamic lighting conditions. Our results demonstrate that our proposed MVS-SLAM approach consistently achieves accurate camera pose estimation and map reconstruction, even in challenging lighting conditions. The semantic module effectively provides contextual information that aids in disambiguating features and maintaining accurate localization.

We performed quantitative analysis by measuring the accuracy of camera pose estimation and comparing it against ground-truth data. The results indicate that our method achieves robust performance across different lighting conditions, with minimal degradation in accuracy. Moreover, qualitative evaluation through visual inspection of the reconstructed maps confirms the ability of our approach to handle lighting variations and produce reliable reconstructions.

4.3.2 | Robustness to occlusions

The robustness of our method to occlusions is facilitated by the integration of semantic information and the utilization of multiview geometry. Occlusions present challenges in traditional SLAM systems, as they hinder feature matching and can lead to inaccurate map reconstruction. In our approach, the semantic module plays a crucial role in handling occlusions by providing contextual information about the scene. By leveraging semantic cues, our system can reason about occluded objects and infer their likely positions, even when they are not directly visible. This semantic understanding aids in maintaining accurate camera pose estimation and reducing the impact of occlusions on the map reconstruction.

To assess the robustness of occlusions, we performed experiments on data sets containing scenes with varying levels of occlusions. Our evaluation results demonstrate that our proposed MVS-SLAM approach exhibits robustness to occlusions and produces accurate reconstructions. We compared our method against state-of-the-art RGBD SLAM approaches and observed superior performance in terms of maintaining accurate pose estimation and producing more complete reconstructions in the presence of occlusions. In summary, our MVS-SLAM approach demonstrates robustness to changes in lighting conditions and occlusions. The integration of semantic information enables the system to reason about the scene, compensate for lighting variations, and handle occlusions effectively. The utilization of MVS techniques further strengthens the system's ability to reconstruct accurate and complete maps under challenging environmental conditions.

We have presented quantitative and qualitative evaluation results in the revised manuscript, showcasing the robustness of our method. These findings support the conclusion that our proposed MVS-SLAM approach offers significant advancements in handling changes in lighting conditions and occlusions, making it well-suited for real-world scenarios where such challenges are prevalent.

5 | SCALABILITY TO DIFFERENT TYPES OF SENSORS AND PLATFORMS

One of the key considerations in the development of our MVS-SLAM system, presented in the manuscript titled “MVS-SLAM: Enhanced multiview geometry for improved semantic RGBD SLAM in dynamic environment,” was its scalability to different types of sensors and platforms. We acknowledge the importance of adaptability and compatibility in enabling the widespread adoption of our approach across various real-world scenarios.

Our MVS-SLAM system was designed with a modular architecture that facilitates seamless integration with different types of sensors. The geometric module, which builds upon the RGBD mode of ORB-SLAM3, is sensor-agnostic and can readily accommodate a range of depth cameras. Whether it is a Microsoft Kinect, Intel RealSense, or a custom-built depth sensor, our system can leverage the depth information provided by these sensors to perform accurate

and robust SLAM. This flexibility allows users to choose the most suitable depth sensor for their specific application requirements.

Moreover, our semantic module, which incorporates the YOLOv7 object detection network, can be adapted to different types of sensors and platforms. While our experiments primarily focused on RGBD data, the semantic module can also leverage RGB information from standard cameras, making it compatible with monocular visual input. By training the DNN on data sets captured by different sensors, we can tailor the semantic module to specific sensor characteristics, environmental conditions, and application domains. This adaptability enhances the system's robustness and enables it to handle diverse sensing modalities.

In terms of platform scalability, our MVS-SLAM system can be seamlessly integrated into various robotic platforms. Whether it is an autonomous ground robot, a flying drone, or a mobile manipulator, the modularity of our approach enables straightforward integration and utilization of our SLAM capabilities. As long as the platform can provide the necessary sensor data (RGB and depth, or RGB-only for the semantic module) and meet the computational requirements, our system can be effectively deployed. This versatility makes our approach applicable to a wide range of robotic applications and enables users to leverage SLAM functionalities in their preferred platforms.

It is important to note that the scalability of our method to different sensors and platforms was a fundamental aspect of our research and development process. We aimed to create a solution that can be readily adapted to diverse sensor configurations and utilized on a variety of robotic platforms. By providing this scalability, we hope to facilitate the adoption of our MVS-SLAM system in real-world applications and encourage its integration into a wide range of robotic systems.

In summary, our MVS-SLAM system exhibits scalability to different types of sensors and platforms. Its modular architecture and adaptability to various sensing modalities enable seamless integration and utilization in diverse robotic applications. We believe that the scalability of our proposed method enhances its practicality and contributes to its potential for broader deployment in the field of SLAM and robotics.

6 | DISCUSSION

In this section, we delve into a comprehensive analysis of the MVS-SLAM framework's distinct components and their collective impact on addressing the intricate challenges posed by dynamic feature points within SLAM. Our approach, an extension of ORB-SLAM3, capitalizes on the integration of three pivotal elements: the object detection thread, the ego-motion estimation module, and the dynamic feature points removal module.

By harnessing the capabilities of a cutting-edge object detection network, the object detection thread infuses semantic awareness into the SLAM process. This semantic prior information, coupled with the Lucas–Kanade method, facilitates the accurate recovery of the initial camera pose. Moreover, our devised strategies within this module act as fail-safes in scenarios that may otherwise lead to ambiguity or uncertainty.

The crux of our solution lies within the dynamic feature points removal module. Operating on a seemingly simple principle, this module distinguishes between dynamic and static points using an ingenious reprojection offset vector-based technique. While seemingly uncomplicated, this approach effectively rids the SLAM system of dynamic points, ensuring a robust tracking and mapping process.

7 | LIMITATIONS AND FUTURE WORK

In light of the promising outcomes showcased by our MVS-SLAM approach in enhancing SLAM accuracy and robustness within dynamic environments, it's imperative to recognize its constraints and chart pathways for future advancements. In this section, we delve into these limitations and their implications for our method's performance and utility.

7.1 | Semantic understanding boundaries

Our methodology hinges on a pretrained object detection network for semantically enriching information. Nonetheless, this approach's grasp is confined to predefined classes of objects within its training data. This might pose challenges when encountering objects or scenes that stray substantially from the learned semantic contexts. To overcome this, augmenting the training data set to encompass a more extensive array of objects and scenarios would empower the system to adeptly tackle novel or less conventional instances.

7.2 | Sensitivity to object detection precision

The accuracy of our approach is intertwined with the precision of its object detection module. Errors or instances of misidentification can result in erroneous semantic labels or an incomplete comprehension of the scene, potentially compromising camera pose estimation and map reconstruction. Elevating the reliability and accuracy of this module should be a focal point to ensure the unfaltering and dependable operation of our MVS-SLAM system.

7.3 | Computational demands

Our approach, weaving together multiview data processing, semantic integration, and real-time map construction, requires substantial computational resources. This might hinder its deployment on platforms with limited resources or applications necessitating stringent latency adherence. To address this, exploring optimization techniques encompassing streamlined data structures and parallel computation methodologies proves pertinent. Additionally, harnessing hardware acceleration avenues

such as graphics processing unit utilization holds the promise of significantly enhancing the computational efficiency of our approach.

7.4 | Scalability across sensor modalities

While meticulously tailored for RGBD sensors, the direct transposition of our MVS-SLAM approach to alternative sensor modalities might encounter challenges. Distinct sensor characteristics including noise profiles, field of view, and measurement precision could mandate custom adaptations or extensions of our methodology. Prospective research avenues should be directed at fabricating sensor-specific modules or investigating sensor fusion strategies to enable the seamless extension of our approach to various sensor configurations.

7.4.1 | Future avenues for exploration

(a) *Exploring diverse deep learning paradigms*

While our current implementation leans on YOLOv7 for semantic extraction, the realm of deep learning harbors a multitude of models and architectures offering unique advantages or superior performance. Unearthing these alternative paradigms stands as an intriguing trajectory for exploration. For instance, probing cutting-edge models, like, EfficientDet, DETR, or Mask R-CNN could potentially yield heightened semantic comprehension and richer feature extraction from RGBD imagery. Likewise, tailoring the network architecture to address the nuances of dynamic environments might bolster both the accuracy and resilience of our system.

(b) *Enriching sensor fusion*

Introducing supplemental sensor modalities into our system emerges as a promising frontier. While RGBD sensors furnish invaluable depth cues, fusing data from diverse sensors can furnish complementary insights and measurements to amplify SLAM efficacy. For instance, melding data from inertial sensors like accelerometers and gyroscopes could shore up motion estimation and counteract sensor drift. Furthermore, amalgamating data from LIDAR or radar sources could bolster resilience against challenging environmental conditions, occlusions, and lighting disparities. The exploration of sensor fusion techniques and the crafting of bespoke fusion frameworks tailored for dynamic RGBD SLAM represent captivating trajectories for future inquiry.

(c) *Dynamic object tracking and cartography*

In the realm of dynamic settings, the accurate tracing and modeling of moving entities carry a profound import for comprehensive scene comprehension. Forthcoming research could pivot towards devising algorithms and frameworks attuned to tracking and integrating dynamic objects within the SLAM paradigm. This entails adept handling of object motion, occlusions, and appearance alterations. By effectively encapsulating the dynamics at play, we stand to yield more precise and consistent mapping outcomes, especially in taxing scenarios.

(d) *Real-time performance augmentation*

While our MVS-SLAM system demonstrably operates in real-time, there's room for further fine-tuning its computational efficiency and memory utilization. Exploring avenues such as network compression, quantization techniques, or capitalizing on hardware acceleration holds the potential to bolster our system's overall performance, rendering it amenable to deployment across resource-constrained platforms or in settings demanding higher computational agility.

8 | CONCLUSION

The novel MVS-SLAM framework presented herein marks a significant leap forward in addressing the intricate challenge of dynamic feature points within the realm of SLAM. Our amalgamation of the object detection thread, ego-motion estimation module, and dynamic feature points removal module yields remarkable advancements in the performance of semantic RGBD SLAM in dynamic environments. Undoubtedly, the experimental results, conducted rigorously on the demanding TUM RGBD data set encompassing sequences with varying dynamic complexities, underscore the supremacy of MVS-SLAM. It outshines existing state-of-the-art SLAM systems in both dynamic and static scenarios, validating its efficacy in surmounting the limitations engendered by dynamic feature points.

As our research journey unfolds, we pinpoint two focal areas necessitating further refinement. The enhancement of image-matching accuracy stands as a paramount objective, ensuring the precision of feature correspondence. Concurrently, improvements in in-depth estimation techniques are essential to heighten overall accuracy. In our subsequent endeavors, we aim to harness the potential of the BEBLID feature point descriptor as an alternative to the prevalent ORB feature points. Additionally, our roadmap involves the adaptation of MVS-SLAM to embrace a binocular strategy, broadening its applicability across diverse contexts.

In summation, the MVS-SLAM framework introduced herein propels the field of SLAM by surmounting the challenges posed by dynamic feature points. Through the strategic integration of advanced components, we elevate the performance of semantic RGBD SLAM in dynamic environments. Our ongoing research efforts are dedicated to the refinement of this framework, both in terms of accuracy and applicability, as we remain committed to driving the advancement of SLAM technology within the realm of robotics.

DATA AVAILABILITY STATEMENT

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

REFERENCES

Baker, S. & Matthews, I. (2004) Lucas–Kanade 20 years on: a unifying framework. *International Journal of Computer Vision*, 56, 221–255.

Bescos, B., Facil, J.M., Civera, J. & Neira, J. (2018) DynaSLAM: tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4), 4076–4083.

Campos, C., Elvira, R., Rodriguez, J.J.G., M. Montiel, J.M. & D. Tardos, J. (2021) Orb-slam3: an accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6), 1874–1890.

Cheng, J., Wang, C. & Meng, M.Q.H. (2020) Robust visual localization in dynamic environments based on sparse motion removal. *IEEE Transactions on Automation Science and Engineering*, 17(2), 658–669.

Cheng, J., Wang, Z., Zhou, H., Li, L. & Yao, J. (2020) DM-SLAM: a feature-based SLAM system for rigid dynamic scenes. *ISPRS International Journal of Geo-Information*, 9(4), 202.

Cui, L. & Ma, C. (2019) SOF-SLAM: a semantic visual SLAM for dynamic environments. *IEEE Access*, 7, 166528–166539.

Dai, W., Zhang, Y., Li, P., Fang, Z. & Scherer, S. (2022) RGB-D SLAM in dynamic environments using point correlations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1), 373–389.

Davison. (2003) Real-time simultaneous localisation and mapping with a single camera. In: *Proceedings of the Ninth IEEE International Conference on Computer Vision*, Vol. 2. France: IEEE, pp. 1403–1410. https://doi.org/10.1109/ICCV.2003.1238654

Endres, F., Hess, J., Sturm, J., Cremers, D. & Burgard, W. (2014) 3-D mapping with an RGB-D camera. *IEEE Transactions on Robotics*, 30(1), 177–187.

Engel, J., Schöps, T. & Cremers, D. (2014) LSD-SLAM: large-scale direct monocular SLAM. In: *Proceedings of the 13th European Conference on Computer Vision—ECCV 2014, Zurich, Switzerland, September 6–12, 2014, Part II*. Springer International Publishing. pp. 834–849.

Fang, B., Mei, G., Yuan, X., Wang, L., Wang, Z. & Wang, J. (2021) Visual SLAM for robot navigation in healthcare facility. *Pattern Recognition*, 113, 107822.

Fischler, M.A. & Bolles, R.C. (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395.

Forster, C., Pizzoli, M. & Scaramuzza, D. (2014) SVO: fast semi-direct monocular visual odometry. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 15–22.

Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Martinez-Gonzalez, P. & Garcia-Rodriguez, J. (2018) A survey on deep learning techniques for image and video semantic segmentation. *Applied Soft Computing*, 70, 41–65.

Kim, D.H. & Kim, J.H. (2016) Effective background model-based RGB-D dense visual odometry in a dynamic environment. *IEEE Transactions on Robotics*, 32(6), 1565–1573.

Li, F., Chen, W., Xu, W., Huang, L., Li, D., Cai, S. et al. (2020) A mobile robot visual SLAM system with enhanced semantics segmentation. *IEEE Access*, 8, 25442–25458.

Li, P., Qin, T., Hu, B., Zhu, F. & Shen, S. (2017) Monocular visual–inertial state estimation for mobile augmented reality. In: *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, pp. 11–21.

Liu, C., Kong, D., Wang, S., Wang, Z., Li, J. & Yin, B. (2021) Deep3D reconstruction: methods, data, and challenges. *Frontiers of Information Technology & Electronic Engineering*, 22(5), 652–672.

Luo, H.L. & Chen, H.K. (2020) Survey of object detection based on deep learning. *Acta Electonica Sinica*, 48(6), 1230.

Mur-Artal, R. & Tardos, J.D. (2017) ORB-SLAM2: an open-source slam system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5), 1255–1262.

Nister, D. (2004) An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6), 756–770.

Qin, T., Li, P. & Shen, S. (2018) Vins-mono: a robust and versatile monocular visual–inertial state estimator. *IEEE Transactions on Robotics*, 34(4), 1004–1020.

Rublee, E., Rabaud, V., Konolige, K. & Bradski, G. (2011) ORB: an efficient alternative to SIFT or SURF. In: *2011 International Conference on Computer Vision*. IEEE, pp. 2564–2571.

Sturm, J., Engelhard, N., Endres, F., Burgard, W. & Cremers, D. (2012) A benchmark for the evaluation of RGB-D SLAM systems. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 573–580.

Tsai, C.F. (2012) Bag-of-words representation in image annotation: a review. *International Scholarly Research Notices*, 2012, 1–9. https://doi.org/10.5402/2012/376804

Wang, C.Y., Bochkovskiy, A. & Liao, H.Y.M. (2023) YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7464–7475.

Wang, R., Wan, W., Wang, Y. & Di, K. (2019) A new RGB-D SLAM method with moving object detection for dynamic indoor scenes. *Remote Sensing*, 11(10), 1143.

Wu, W., Guo, L., Gao, H., You, Z., Liu, Y. & Chen, Z. (2022) YOLO-SLAM: a semantic SLAM system towards dynamic environment with geometric constraint. *Neural Computing and Applications*, 34, 6011–6026.

You, Y., Wei, P., Cai, J., Huang, W., Kang, R. & Liu, H. (2022) MISD-SLAM: multimodal semantic SLAM for dynamic environments. *Wireless Communications and Mobile Computing*, 2022, 1–13.

Yu, C., Liu, Z., Liu, X.J., Xie, F., Yang, Y., Wei, Q. et al. (2018) DS-SLAM: a semantic visual SLAM towards dynamic environments. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 1168–1174.

Zhang, Z. (2012) Microsoft Kinect sensor and its effect. *IEEE Multimedia*, 19(2), 4–10.

Zhang, J., Shi, C. & Wang, Y. (2020) SLAM method based on visual features in dynamic scene. *Computer Engineering*, 46, 95–102.

Zhao, Y., Yan, L., Chen, Y., Dai, J. & Liu, Y. (2021) Robust and efficient trajectory replanning based on guiding path for quadrotor fast autonomous flight. *Remote Sensing*, 13(5), 972.

Zhao, Z.Q., Zheng, P., Xu, S.T. & Wu, X. (2019) Object detection with deep learning: a review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11), 3212–3232.

Zhong, F., Wang, S., Zhang, Z. & Wang, Y. (2018) Detect-SLAM: making object detection and SLAM mutually beneficial. In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1001–1010.

How to cite this article: Islam, Q.U., Ibrahim, H., Chin, P.K., Lim, K. & Abdullah, M.Z. (2023) MVS-SLAM: Enhanced multiview geometry for improved semantic RGBD SLAM in dynamic environment. *Journal of Field Robotics*, 1–22. https://doi.org/10.1002/rob.22248

View publication stats