ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Visual SLAM Framework with Culling Strategy Enhancement for Dynamic Object Detection

Pattanapong Saeyong[1] and Kittikhun Thongpull[2]

ABSTRACT

Visual Simultaneous Localization and Mapping (visual SLAM or vSLAM) enables robots to navigate and perform complex tasks in an unfamiliar environment. Most visual SLAM techniques can operate effectively under a static environment where objects are stationary. However, in practical applications, the environment often consists of moving objects and is dynamic. Visual SLAM methods designed for the static environment do not perform well in the dynamic one. In this paper, we propose an additional process to enhance the performance of visual SLAM for a dynamic environment. Our proposed visual SLAM system for dynamic circumstances, based on ORB-SLAM2, combines the capabilities of dynamic object detection and background inpainting to reduce the effect of dynamic objects. The system can detect moving objects using both semantic segmentation and LK optical flow with the epipolar constraint method, and the localization accuracy can be improved in dynamic scenarios. Having a specific scene map allows inpainting the obscured background from such dynamic objects utilizing static information that occurs at previous views. Eventually, a semantic octomap is built, which could be applied for navigation and high-level tasks. The experiment was carried out on the TUM RGB-D dataset and real-world environment and implemented on Robot Operating System (ROS). The experimental results show that the Absolute Trajectory Error (ATE) reduce up to 98.03% compared with standard visual SLAM baselines. It can fully demonstrate that the proposed object detection process can detect movable objects and reduce the impact of dynamic objects in visual SLAM.

Article information:

Keywords: Visual SLAM, Dynamic Environment, Semantic Segmentation, Optical Flow, Octomap

Article history:
Received: March 15, 2023
Revised: May 1, 2023
Accepted: August 3, 2023
Published: September 2, 2023
(Online)

DOI: 10.37936/ecti-cit.2023173.251998

1. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) is used to localize an autonomous mobile robot in a GPS-denied environment or unknown scenario. SLAM constructs a consistent map of the surrounding environment through collected data from various sensors [1]. The type of information used in the SLAM computation process can be divided into two techniques. First, Laser SLAM [2], which obtains environment information from a lidar sensor. Second, visual SLAM [3] that occupies vision-based sensors, e.g., RGB camera. Visual SLAM has been expanded in the recent development of SLAM technology because of its cost-efficient and rich information for complex navigation tasks. For example, the 3D mapping process requires direct depth information, which can simply be obtained from an RGB-D camera accompanied by a visual SLAM task [4]. However, the conventional visual SLAM approaches, such as PTAM [5], ORB-SLAM [6], and ORB-SLAM2 [7] that have been used a lot in visual SLAM field have achieved promising performance only in static circumstances which are rarely found in practical applications. The issue is still active in research as it is essential for real-world applications.

Aiming at the issue of the dynamic environment, deep learning-based approaches are utilized to identify dynamic objects by several researchers [8-10] to

[1,2] The authors are with Department of Electrical Engineering Faculty of Engineering, Prince of Songkla University, Thailand, E-mail: pattanapong1236@gmail.com and kittikhun.t@psu.ac.th
[2] The corresponding author: kittikhun.t@psu.ac.th

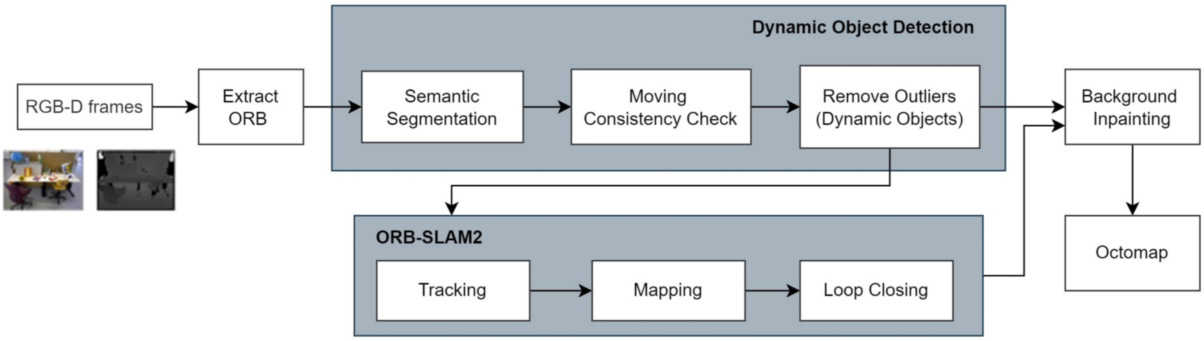


Fig. 1: *The system framework of our proposed algorithm.*

mitigate the uncertainty environment information. In practice, there is still ambiguity in some categories of objects between dynamic and static objects, which deteriorates the identification performance. Therefore, the deep learning-based method should be expanded by merging the geometrical information [11-13] to understand the surrounding map update within a dynamic environment.

Thus, in this paper, we propose a developed visual SLAM framework to cope with the problem of dynamic environment. The system is based on the ORB-SLAM2 [7] algorithm and implemented on Robot Operating System (ROS) [14]. We combine semantic segmentation with the Lucas-Kanade (LK) optical flow method [15] and epipolar constraint [16] to detect dynamic objects and occlude the detected area from the mapping information. Finally, the system generates a semantic octomap to apply for high-level tasks. The main contributions of our work are summarized as follows:

- We propose a dynamic visual SLAM based on an RGB-D sensor that combines an ICNet [17] network and LK optical flow with epipolar constraint to handle dynamic objects on pose estimation, along with a background inpainting approach and generating an octomap.
- We present the strategies of judgment and culling for potentially dynamic feature points to increase accurate dynamic detection.
- We evaluate the performance of our proposed method on public RGB-D TUM datasets and real-world scenarios and achieve better localization performance in high dynamic environment compared to ORB-SLAM2.

The remaining parts of the paper are organized as follows: Section 2 discusses the related works. Then, Section 3 presents the proposed method's description. Section 4 explains the experimental results on public datasets. Finally, a brief conclusion is summarized in Section 5.

2. RELATED WORKS

The type of SLAM frameworks can classify according to the data acquisition method into two significant groups: laser SLAM [2] and visual SLAM [3]. Laser SLAM uses lidar as the sensor to acquire environmental information for the algorithm. For example, Google's Cartographer [18] can perform real-time SLAM with good loop closure results.

Visual SLAM refers to the SLAM technique that utilizes the camera, either monocular, stereo, or RGB-D camera, to create a map [19]. In the past, many visual SLAM algorithms that can achieve well in static circumstances during the experiment, such as PTAM [5], ORB-SLAM [6], and LSD-SLAM [20] were invented. ORB-SLAM2 [7] algorithm, mainly, has been used commonly in visual SLAM research, which provides a complete SLAM system with the ability of map reuse, loop closing, and relocalization to improve the efficiency of the SLAM process.

On the other hand, the dynamic visual SLAM is the extension of the SLAM algorithm to cope with non-stationary properties in an environment in which dynamic objects are examined and regarded as outliers. Then, the identified dynamic features will be discarded, and only the remaining static feature will be employed to calculate the camera position and attitude for map updating and localization process. Recently, methods of combining deep learning with visual SLAM for dynamic object detection have been used for dynamic SLAM. The detection of dynamic objects in traditional mathematical models through geometric information is introduced in Section 2.1, and the dynamic SLAM systems based on deep learning are demonstrated in Section 2.2.

2.1 Dynamic Visual SLAM Based on Traditional Models

Several techniques based on geometric methods have been applied to manipulate dynamic scenes in visual SLAM [21]. The studies [22], and [23] applied multibody Structure-from-Motion (SfM) to deal with dynamic environment. Zou *et al.* [24] proposed Collaborative Visual SLAM or CoSLAM in dynamic environment with multiple cameras. Li *et al.* [25] applied the static weighting method that calculates the likelihood of depth edge points being part of the static environment, which is integrated into the

IAICP method. Sun *et al.* [26] integrated motion removal into the front end of RGB-D SLAM that acts as a pre-processing stage to filter out the moving or dynamic objects; however, this method is limited to scenarios with many moving objects. The LK optical flow [15] approach, which is the apparent motion of brightness patterns in the image, is also used to distinguish and eliminate the dynamic feature point. For example, Cheng *et al.* [27] proposed monocular visual SLAM, which uses the LK optical flow technique and polar geometric constraints to filter out feature points of dynamic targets. Wang *et al.* [28] combined fundamental matrix constraint and depth clustering algorithm for RGB-D SLAM to eliminate the moving feature point.

2.2 Dynamic Visual SLAM Based on Deep Learning

Recently, with the advent of the deep learning network and the development of semantic segmentation to identify and classify objects with superior performance compared to traditional methods. Therefore, some visual SLAM systems are integrated with deep learning networks to improve the efficiency of SLAM systems. DynaSLAM [11] proposed the visual SLAM combined with the Mask R-CNN [29] and multiple-view geometry to filter out dynamic objects. However, the multiple-view geometry method takes much time to process. Long *et al.* [30] used DynaSLAM as a lightweight pose estimation and proposed PSPNet [31] network to obtain a pixel-wise semantic segmentation and combined it with the optimal error compensation homologous matrix to improve the system robustness. Besides, the system applied a reverse ant colony strategy to decrease the time consumption in the multiple-view geometry process. Yu *et al.* [12] present the SLAM named DS-SLAM, which applied the Segnet [32] network to provide semantic information and combine it with the epipolar constraint algorithm to assume that the point features are static or dynamic. In addition, the system could build a dense semantic octo-tree map for high-level tasks. Ran *et al.* [13] also used the Segnet network to accomplish semantic segmentation and integrate it with a Bayesian update method. Cheng *et al.* [33] proposed DM-SLAM, which combines instance segmentation with optical flow information and presents strategies to detect dynamic feature points for RGB-D, stereo, and monocular cameras to handle wrong data associations during the SLAM algorithm.

Currently, the RGB-D camera technology can provide both depth (D) and color (RGB) data as the output simultaneously. Therefore, the visual SLAM can utilize depth information with deep learning to handle dynamic scenarios. Cui *et al.* [34] presented SDF-SLAM or Semantic Depth Filter SLAM, which utilizes the semantic information with depth filter to identify whether a 3D map point is dynamic or static. PLD-SLAM is proposed by Zhang *et al.* [35], which combines deep learning with the K-Mean clustering of depth information in the segmentation area. Furthermore, the system utilizes point and line features to calculate camera pose in the SLAM process.

We tackle the camera pose localization issue in dynamic circumstances. Our proposed method combines semantic segmentation network and epipolar constraint [16] to address the dynamic environment problem. Compared with other methods, our proposed method embeds the judgment and culling step for classifying the feature point as it is a dynamic point or a static point for the objects that cannot move by themselves, but there is a potentially movable property. The overall process was expected to yield improvement in localization accuracy.

3. SYSTEM DESCRIPTION

In this section, the details regarding our proposed visual SLAM will be described thoroughly. The section is divided into five parts. First, the overview of our system is introduced. Second, the details of the semantic segmentation algorithm used in the system are explained. The third part describes the LK optical flow [15] and epipolar constraint [16] for the geometric method that we adopt to distinguish the motion feature in the image. Next, Dynamic points will be culled and discarded in the outlier rejection state. Finally, the background inpainting process and octomap are presented.

3.1 Overview

ORB-SLAM2 [7], the famous featured-based SLAM algorithm, has an effective performance constrained in assuming static situations from indoor to outdoor environment. From the promising performance of CNN (Convolutional Neural Network), we have adopted CNN capabilities and the geometrical method in ORB-SLAM2 for practical applications in which the surrounding environment consist of mobile entities and are dynamic.

Fig. 1 shows the overview of our proposed system, which aims to improve the detection of dynamic objects and create a semantic octomap based on ORB-SLAM2 [7]. The system obtains an RGB-D frame from an RGB-D camera, which provides color image and depth image as the input. First, The RGB channels are processed by semantic segmentation through ICNet [17] for preliminary discrimination of the moving and static contents. The system applies the ICNet [17] with real-time processing to reduce the semantic segmentation delay. Then, we perform the Moving Consistent Check based on the optical flow [15] and epipolar constraint [16] method to label new dynamic objects that were not movable in the semantic segmentation stage. The potentially dynamic and static features will be further considered and classified with judgment strategies in the Remove Outliers step to conclude whether candidate features were dynamic removal points. After the detection process, associated ORB feature points of moving objects have been rejected from the map database. These ORB feature points of stationary content will be applied in the Tracking and Mapping stage of ORB-SLAM2 [7]. Finally, the obscured background will be applied over the region of the detected object; then the Background Inpainting will generate inpainted images for the Octomap process.

3.2 Semantic Segmentation Network

In order to detect moving objects in dynamic conditions, we adopt a pixel-level semantic segmentation of the images. This system utilizes ICNet [17] to obtain semantic segmentation labels. The ICNet was trained on the PASCAL VOC dataset [36]. We determine the potentially dynamic types of objects into three categories. The first is an *active dynamic object*, which can move intrinsically. In addition to organisms such as humans and animals, the movable vehicle is also defined as an active dynamic object type. Then, objects that have a chance to move but cannot move by themselves will be considered as *passive dynamic object.* Finally, static objects or backgrounds will not be identified by the segmentation and are defined as *static object.* Fig. 2 shows the semantic segmentation results that correctly identify the person and chair as a movable object.

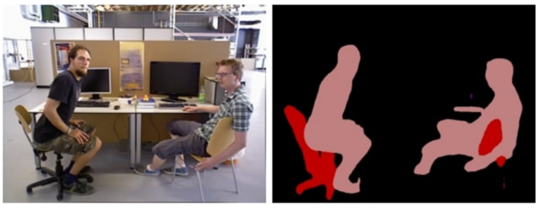


Fig.2: *Visualizing semantic segmentation result.*

Some deep learning-based algorithms are used in several visual SLAM systems. Each model is used to implement for the segmentation of dynamic content. For example, DS-SLAM [12] adopts the Segnet [32] network, and PSPNet-SLAM [30] applies the PSPNet [31] network. One of the challenges of visual SLAM is real-time processing; a promising architecture for fast semantic segmentation is Image Cascade Network or ICNet [17], which is proposed by Zhao *et al.* This model is a CNN-based semantic segmentation that can provide results at a low computational cost or achieve real-time semantic segmentation. ICNet takes cascade image inputs (i.e., low-, medium- and high-resolution images), adopts a cascade feature fusion unit, and is trained with cascade label guidance. The ICNet network architecture is shown in Fig. 3.

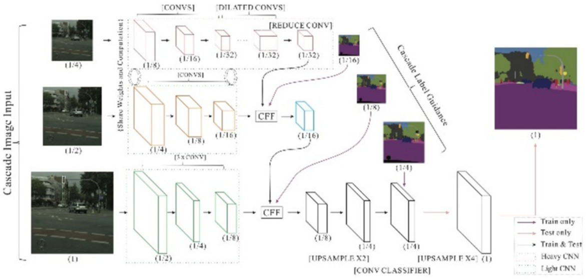


Fig.3: *ICNet network architecture.*

Zhao *et al.* [17] experimented with mIoU performance and inference time on the test set of Cityscapes between ICNet and other methods, which can be presented in Table 1. ICNet method yields mIoU 69.5% less than mIoU 81.2% of PSPNet method and more than mIoU 51.0% of Segnet. Although the mIoU performance of the ICNet is worse than PSPNet method, it can provide real-time processing with the fastest frame rate at 30.3 fps.

Table 1: *mIoU and inference time comparison between ICNet and other methods.*

Method	mIoU (%)	Time (ms)	Frame (fps)
Segnet	51.0	60	167
PSPNet	81.2	1288	0.78
ICNet	69.5	33	30.3

3.3 Moving Consistency Check

Active dynamic object obtained from Section 3.2 will be defined as moving objects and discarded in Section 3.4. However, the passive dynamic object is generally immovable unless it is moved by external influence. In the first step, the feature points of a passive dynamic object from the semantic segmentation stage will be considered with the pyramid LK optical flow [15] method to interpret the mobility. First, we calculate the optical flow pyramid to obtain the matched feature points from a previous frame to the current one. The matched feature points are investigated to determine whether the matched pair is too close to the border of the figure or the pixel difference of the 3×3 image block at the center of the matched pair is too large; the matched feature points will be rejected. Then, the epipolar constraint [16] will be proposed. The fundamental matrix [37] is found using RANSAC [38]. The next step is to compute the epipolar line in the present frame utilizing the fundamental matrix. Eventually, The system estimates whether the displacement from a matched feature points to its consistent epipolar line is below the threshold. The matched point that the distance is more than the threshold will be discarded.

The epipolar constraint is used to classify the potentially dynamic feature points. The matrix F, known as the fundamental matrix (3 × 3 matrix), which was computed from the TUM dataset, is helpful in computing the epipolar lines associated with the feature points in the last frame and the corresponding points in the current frame, which can create a compact mathematical model expressed as follows:

$$q_i^T l = q_i^T F \; p_i = 0 \tag{1}$$

where p_i and q_i represent the matched feature points in the previous frame and the current frame, respectively. l is the epipolar line.

The epipolar constraint model is shown in Fig. 4, $o_p - x_p y_p z_p$ and $o_q - x_q y_q z_q$ indicate the coordinate system in the previous frame and the current frame, respectively. The epipolar line l in the current frame corresponds to feature points p_i in the reference frame. In the present image I_2, the projection of p has to be placed on the epipolar line l, which is described in Eq. 1. Normally, if the point is static, the displacement between q_i and epipolar line l will be close to 0.

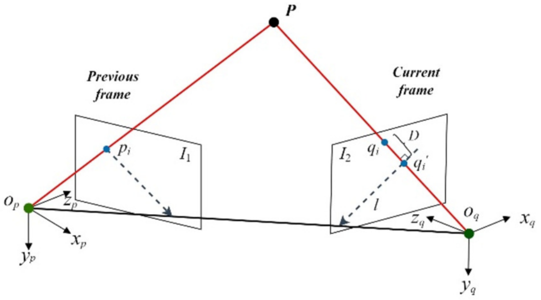


Fig.4: *Epipolar constraint diagram (q_i is the desired point that corresponds to p_i in the previous frame, and q_i is the actual measured point. So, D is the distance between the actual point and the epipolar line).*

Let the homogeneous feature point be $P = [x_i, y_i, 1]^T$, the projected feature points into the previous image $p_i = [x_i^p, y_i^p]^T$, and its corresponding feature point $q_i = [x_i^q, y_i^q]^T$, where x, y are the coordinate value in the image frame. Each feature point lies on the epipolar line l, which is denoted as $[A, B, C]^T$, and can assume the line equation of the epipolar line is $Ax + By + C = 0$. Therefore, the distance between the epipolar line and its matching point from an image is calculated as:

$$D = \frac{|Ax_i^q + By_i^q + C|}{\sqrt{A^2 + B^2}} = \frac{|q_i^T F p_i|}{\sqrt{A^2 + B^2}} \tag{2}$$

where D represents the distance and $|\cdot|$ is the vector norm. Usually, supposing the homogeneous feature point is static. In that case, the projected feature point into the current frame will lie on the epipolar line (l) according to the epipolar geometry model [16]. The distance is calculated from the feature point to the epipolar line to determine whether the distance is greater than the preset threshold value. The threshold value is 1.0, according to Zhang *et al.* [35]. If this value is adjusted to a higher value, the slight to moderate movement of the object will not be detected. Thus, SLAM efficiency may decrease in low dynamic environment. Conversely, a lower threshold value will cause a lot of slight motion detections in the SLAM process. Feature point was eliminated excessively making positioning inaccurate. If the distance surpasses the preset distance value, the potentially dynamic feature point will be considered a dynamic feature point and eliminated from the whole point before accessing the tracking process.

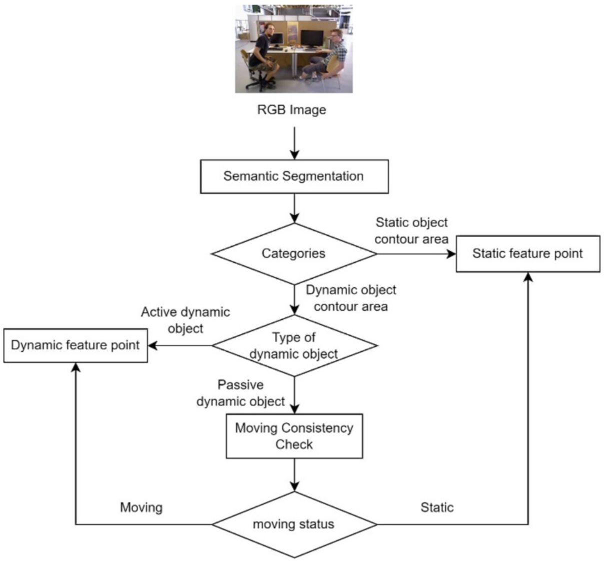


Fig.5: *The flowchart of judgment and culling for potentially dynamic feature points.*

3.4 Remove Outliers

After separating the three categories from Section 3.2 semantic segmentation: *active dynamic object, passive dynamic object,* and *static object*, all of the feature points in the image will be judged by combining these semantic categories with the geometric method from Section 3.3. This step is to classify the feature point, whether it is the dynamic point, which has been removed, or a static point to be used in the SLAM process. Fig. 5 shows the flowchart of judgment and culling for potentially dynamic feature points. The RGB image is processed in the semantic segmentation step to classify categories from predefined classes. There are two main categories of object contour area: Static object and Dynamic object. The feature points that fall in a static object contour area will be arranged in static points. The dynamic object type can be divided into an active dynamic object and a passive dynamic object. If a feature point belongs to an active dynamic object, it is considered an actual dynamic feature point. In another way, if the object is identified as a passive dynamic object, all feature points falling in this contour area will be geometrically examined by the Moving Consistency

Table 2: *Result of metrics Absolute Trajectory Error (ATE) [m] in meters.*

Sequences	ORB-SLAM2				Proposed Method				Improvement (%)			
	RMSE	Mean	Median	SD	RMSE	Mean	Median	SD	RMSE	Mean	Median	SD
fr3_s_static	0.0093	0.0083	0.0075	0.0042	0.0071	0.0062	0.0034	0.0034	23.66	25.30	26.67	19.05
fr3_s_xyz	0.0173	0.0152	0.0136	0.0082	0.0101	0.0087	0.0078	0.0049	41.62	42.76	42.65	40.24
fr3_s_half	0.0525	0.0484	0.0163	0.0204	0.0168	0.0141	0.0117	0.0112	68.00	70.87	74.43	45.10
fr3_w_static	0.1598	0.1347	0.1177	0.0860	0.0098	0.0074	0.0060	0.0064	93.87	94.51	94.90	92.56
fr3_w_xyz	0.7247	0.5954	0.5181	0.4131	0.0143	0.0125	0.0115	0.0069	98.03	97.90	97.78	98.33
fr3_w_half	0.4120	0.3640	0.3326	0.1930	0.0272	0.0264	0.0228	0.0155	93.40	92.75	93.14	91.97
fr3_w_rpy	0.8005	0.7021	0.6385	0.3845	0.1469	0.1270	0.1025	0.0738	81.65	81.91	83.95	80.81

Table 3: *Result of metrics Translational Drift (RPE) [m/s] in meters/second.*

Sequences	ORB-SLAM2				Proposed Method				Improvement (%)			
	RMSE	Mean	Median	SD	RMSE	Mean	Median	SD	RMSE	Mean	Median	SD
fr3_s_static	0.0103	0.0092	0.0084	0.0046	0.0081	0.0071	0.0065	0.0037	21.36	22.83	22.62	19.57
fr3_s_xyz	0.0146	0.0126	0.0113	0.0073	0.0128	0.0111	0.0098	0.0063	12.33	11.90	13.27	13.70
fr3_s_half	0.0379	0.0277	0.0199	0.0258	0.0231	0.0173	0.0136	0.0154	39.05	37.55	31.66	40.31
fr3_w_static	0.1024	0.0464	0.0167	0.0913	0.0135	0.0105	0.0087	0.0085	86.82	77.37	47.90	90.69
fr3_w_xyz	0.3945	0.2929	0.2141	0.2644	0.0196	0.0169	0.0149	0.0099	95.03	94.23	93.04	96.26
fr3_w_half	0.3556	0.1970	0.0503	0.2960	0.0299	0.0254	0.0228	0.0157	91.59	87.11	54.67	94.70
fr3_w_rpy	0.3916	0.2732	0.1353	0.2806	0.0721	0.0502	0.0340	0.0518	81.59	81.63	74.87	81.54

Table 4: *Result of metrics Rotational Drift (RPE) [deg/s] in degree/second.*

Sequences	ORB-SLAM2				Proposed Method				Improvement (%)			
	RMSE	Mean	Median	SD	RMSE	Mean	Median	SD	RMSE	Mean	Median	SD
fr3_s_static	0.2974	0.2705	0.0045	0.1236	0.2749	0.2463	0.0040	0.1222	7.57	8.95	11.11	1.13
fr3_s_xyz	0.4746	0.4101	0.0064	0.2388	0.4961	0.4169	0.0061	0.2687	-4.53	-1.66	4.69	-12.52
fr3_s_half	0.9164	0.7679	0.0112	0.5000	0.6651	0.5632	0.0085	0.3537	27.42	26.66	24.11	29.26
fr3_w_static	1.7723	0.8584	0.0063	1.5505	0.3149	0.2656	0.0040	0.1692	82.23	69.06	36.51	89.09
fr3_w_xyz	7.5135	5.6312	0.0694	4.9741	0.5869	0.4624	0.0067	0.3614	92.19	91.79	90.35	92.73
fr3_w_half	7.4235	4.1707	0.0213	6.1412	0.8207	0.7088	0.0108	0.4138	88.94	83.01	49.30	93.26
fr3_w_rpy	7.6974	5.4260	0.0450	5.4593	1.4539	1.0284	0.0129	1.0276	81.11	81.05	71.33	81.18

Table 5: *Comparison results of Absolute Trajectory Error (ATE) [m] for our system against other algorithms.*

Sequences	DS-SLAM	Dyna-SLAM	DM-SLAM	Proposed Method
fr3_s_static	0.0065	0.0064	**0.0063**	0.0071
fr3_s_xyz	-	0.0130	-	**0.0101**
fr3_s_half	**0.0148**	0.0191	0.0178	0.0168
fr3_w_static	0.0081	0.0080	**0.0079**	0.0098
fr3_w_xyz	0.0247	0.0158	0.0148	**0.0143**
fr3_w_half	0.0303	0.0274	0.0274	**0.0272**
fr3_w_rpy	0.4442	0.0402	**0.0328**	0.1469

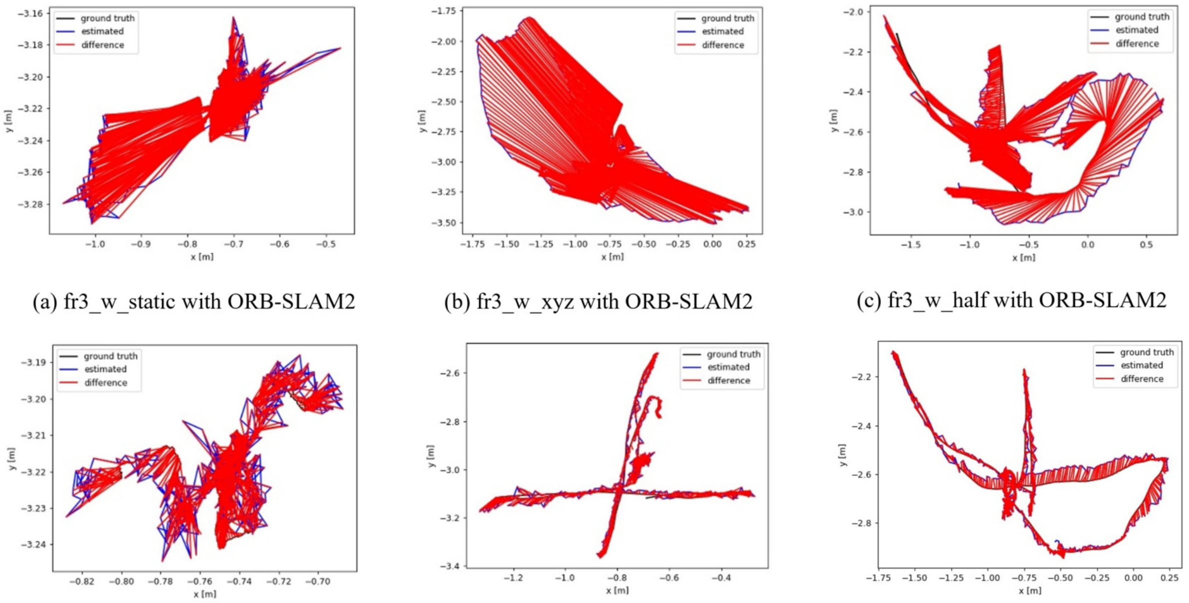
Check step. The feature points that are located in the segmented object determined to be a moving object, will be discarded. Otherwise, the object and its feature points are considered to be static.

3.5 Background Inpainting and Octomap

For every disposed dynamic feature point, we perform inpainting the occluded background with the estimated static content. Since we know the position of the previous and current frame, we project and synthesize a set of previous keyframes into the removed dynamic content of the current frame. Then, the local point cloud will be kept and transformed from these inpainted frames to the real-world coordinate, creating a global Octomap [39]. Octomap implements a 3D occupancy grid mapping approach, allowing data structures and mapping algorithms. The approach is based on the structure of the octree [39]. Octomap is flexible, concise, updatable, and employed easily for navigation.

4. EXPERIMENT AND RESULT

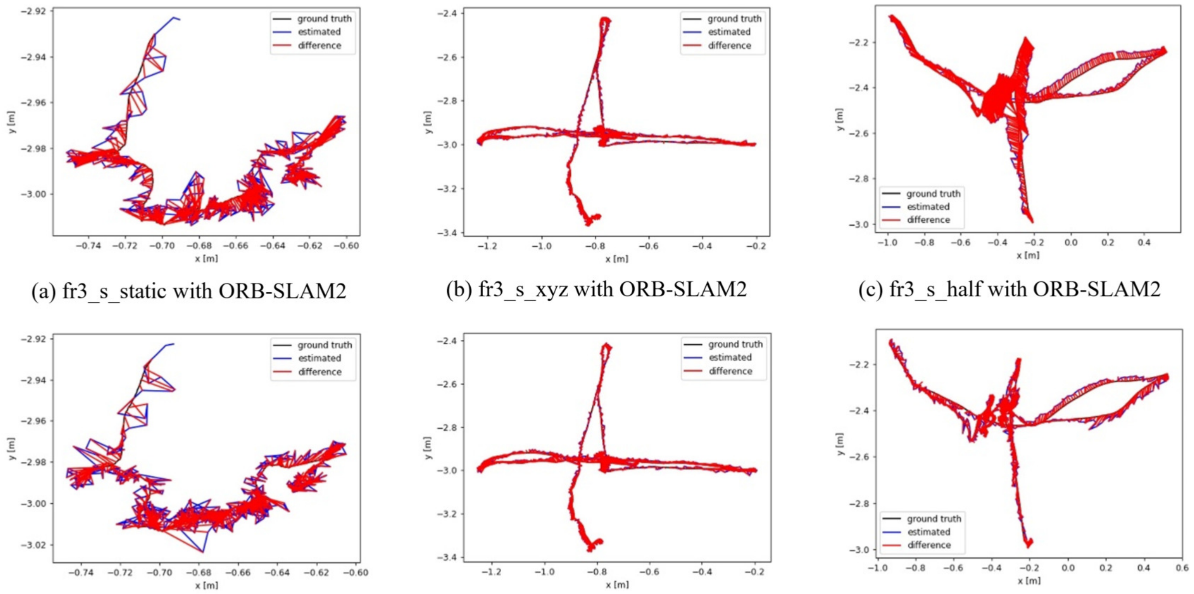
In this section, the proposed visual SLAM system has been evaluated using the TUM RGB-D dataset [40] to investigate its performance. The experiment are performed on a PC with IntelCorei7-8665U CPU, Mesa intel(r) UHD Graphics 620 (WHL GT2), and RAM 8 GB. The system environment was the Ubuntu 20.04 operating system, and the experiment was the ROS implementation.



(a) fr3_w_static with ORB-SLAM2 (b) fr3_w_xyz with ORB-SLAM2 (c) fr3_w_half with ORB-SLAM2

(d) fr3_w_static with the proposed method (e) fr3_w_xyz with the proposed method (f) fr3_w_half with the proposed method

Fig.6: *Plot of ATE[m] (trajectory) for high dynamic environment: fr3_w_static, fr3_w_xyz, fr3_w_half.* ***(a-c)*** *the experiments executed with ORB-SLAM2;* ***(d-f)*** *the experiments executed with our proposed method.*



(a) fr3_s_static with ORB-SLAM2 (b) fr3_s_xyz with ORB-SLAM2 (c) fr3_s_half with ORB-SLAM2

(d) fr3_s_static with the proposed method (e) fr3_s_xyz with the proposed method (f) fr3_s_half with the proposed method

Fig.7: *Plot of ATE[m] (trajectory) for high dynamic environment: fr3_s_static, fr3_s_xyz, fr3_s_half.* ***(a-c)*** *the experiments executed with ORB-SLAM2;* ***(d-f)*** *the experiments executed with our proposed method.*

4.1 TUM RGB-D Dataset

The TUM RGB-D dataset [40] contains numerous dynamic scene sequences recorded by Microsoft Kinect sensors at 30 fps with 640 × 480 resolution. The dataset consists of two scenarios aiming to characterize the dynamic situation. The first scenario captures an event of two people sitting along with talking and making a gesture named “s” for *sitting*. So, we define this sequence as *low dynamic environment*. The second scenario is determined as *high dynamic environment*, where two people walk around and sit down at the desk in walking sequence. This scenario is denoted as the letter “w”. For both types of sequences, *sitting* and *walking*, there are four camera motion types: *static*, the camera is kept static manually; *xyz*, camera movement along the *xyz* axis; *half sphere (half)*, the camera moves according to the path of a 1-meter diameter half sphere; and rpy, the camera rotates over *rpy* axis (roll, pitch and yaw axes).

4.2 Evaluation of TUM RGB-D Dataset

We utilize the metric of Absolute Trajectory Error (ATE) [40] for quantitative measuring the performance of visual SLAM systems. Relative Pose Error (RPE) [40] is well-suited for measuring the drift of a visual odometry system, which measures in the translational and rotational drift terms. We experimented by comparing our proposed method with RGB-D ORB-SLAM2. In Table 2- 4, the first column is the name of the sequence name, and the initials fr3_w_half represent the freiburg3_walking_half sphere sequence. For statistical value comparison, we present the value of Root Mean Squared Error (RMSE), Mean Error, Median Error, and Standard Deviation error (SD). Root Mean Squared Error (RMSE) describes the difference between the actual value and the estimated value; Mean Error and Median Error show the average and medium levels of estimated error, respectively; Standard Deviation error (SD) represents a measure of how dispersed the estimated error is concerning the mean error. From the mentioned measured values, the robustness and stability of the visual SLAM system can be referenced from RMSE and SD values. The performance improvement of statistical values has been computed in comparison to ORB-SLAM2 given in the last column. The improvement formula is computed by:

$$I = \left(1 - \frac{P}{T}\right) \times 100 \tag{3}$$

where P represents the value of our proposed method, T denotes the value of the original ORB-SLAM2, and I represents the improvement percentage.

As the experimental results of seven test sequences shown in Table 2-4, our proposed method can significantly decrease the Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) of both translational and rotational errors in image sequences, exceptionally *high dynamic environment*. The improvement is enhanced in the *high dynamic environment*, the improved ATE value of RMSE and SD were 98.03% and 98.33%, respectively. Likewise, the Relative Pose Error (RPE) is reduced regarding the ATE enhancement. The results show that our proposed method has the capability of performance enhancement compared with ORB-SLAM2 in high dynamic scenarios. However, for the *low dynamic environment*, the results of three sequences were slightly improved from the results obtained by ORB-SLAM2 and became worse in fr3_s_xyz or freiburg3_sitting_xyz sequence for Rotational Drift (RPE). The primary reason may be the effect of the low dynamic movements typically occurring intermittently in the sequences. Also, moving objects that always appear motionless in some frames have high performance degradation impact. In addition, the tracked feature points locate at a greater distance than those associated with dynamic objects. Therefore, the original ORB-SLAM2 can manipulate well and achieve better accuracy. According to the results, the proposed system's performance deterioration may occur in stationary and *low dynamic environment*.

As shown in Table 5, The comparison results of Absolute Trajectory Error (ATE) indicate that our approach surpassed other methods in freiburg3_sitting_xyz, freiburg3_walking_xyz, and freiburg3_walking_half sphere sequence. The results indicate that our proposed approach can enhance the robustness and consistency of the SLAM process in *high dynamic environment*. Regarding other sequences, our method's performance is almost close to the method list in every sequences. However, the error of the freiburg3_walking_rpy is that much worse than Dyna-SLAM and DM-SLAM because the angular velocity of camera movement in this sequence is relatively fast. So, the tracking process has disappeared, and the estimated localization has a significant error with the ground truth for a while. In addition, the processing performance of a desktop computer is also necessary. Processing in the semantic segmentation step takes a lot of time consumption and processing resources. As a result, the tracking process lost at certain intervals. If the experiment was carried out on a higher-performance PC, the capability of calculating and processing would be improved. The experimental results are explicitly enhanced in the freiburg3_walking_rpy sequence.

Fig. 6 and 7 display the motion trajectory or ATE graph of ORB-SLAM2 and our proposed method compared to the ground truth trajectory, along with showing the difference with the estimated trajectory. Fig. 6 shows the ATE curve graph for *high dynamic environment*. The generated trajectory is similar to the ground truth path, that is, the accuracy enhancement in the proposed method because of disposing of the influence in dynamic scenes. Fig. 7 shows the ATE curve graph for *low dynamic environment.* It can be seen that the original ORB-SLAM2 perform well in these cases. The error is not reduced compared to the *high dynamic environment.*

5. CONCLUSIONS

In this paper, a proposed visual SLAM performance enhancement technique for the dynamic environment is developed. This system is based on ORB-SLAM2 combined with the capabilities of dynamic object detection, which is an additional front-end stage to ORB-SLAM2 using semantic segmentation, optical flow method, and epipolar constraint for filtering out and reducing the effect of dynamic objects. In addition, we apply the judgment and culling for potentially dynamic feature points to increase the precision of dynamic object detection. Then, a static map allows inpainting the removed dynamic background. Finally, the synthetic frames from the inpainting stage will be generated to a semantic octomap. The experiments were carried out on the TUM RGB-D dataset and a real-world environment based on Robot Operating System (ROS). The result shows that the proposed object detection process can accurately detect movable objects and remove the feature point from the map database. The experimental results show that the Absolute Trajectory Error (ATE) is reduced up to 98.03% compared with ORB-SLAM2. It can be concluded entirely that the proposed object detection process can detect movable objects and improve visual SLAM performance in a dynamic environment.

References

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *IEEE Robotics Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[2] X. Niu, C. Zhang, S. Fu, and W. Zhang, "Research on the development of 3d laser slam technology," in *2021 IEEE 4th International Conference on Big Data and Artificial Intelligence (BDAI)*, pp. 181–185, 2021.

[3] J. Fuentes-Pacheco, J. R. Ascencio, and J. M. Rendon-Mancha, "Visual simultaneous localization and mapping: a survey," *Artificial Intelligence Review*, vol. 43, pp. 55–81, 2012.

[4] H. Jo, S. Jo, H. M. Cho, and E. Kim, "Efficient 3d mapping with rgb-d camera based on distance dependent update," in *2016 16th International Conference on Control, Automation and Systems (ICCAS)*, pp. 873–875, 2016.

[5] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 225–234, 2007.

[6] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[7] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[8] J. Cheng, Y. Sun, and M. Q.-H. Meng, "A dense semantic mapping system based on crf-rnn network," in *2017 18th International Conference on Advanced Robotics (ICAR)*, pp. 589–594, 2017.

[9] Y. Sun, W. Zuo, and M. Liu, "Rtfnet: Rgb-thermal fusion network for semantic segmentation of urban scenes," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2576–2583, 2019.

[10] C. Zhao, L. Sun, and R. Stolkin, "A fully end-to-end deep learning approach for real-time simultaneous 3d reconstruction and material recognition," in *2017 18th International Conference on Advanced Robotics (ICAR)*, pp. 75–82, 2017.

[11] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "Dynaslam: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4076–4083, 2018.

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

[13] T. Ran, L. Yuan, J. Zhang, D. Tang, and L. He, "Rs-slam: A robust semantic slam in dynamic environments based on rgb-d sensor," *IEEE Sensors Journal*, vol. 21, no. 18, pp. 20657–20664, 2021.

[14] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler and A. Ng, "Ros: an open-source robot operating system," *IEEE International Conference on Robotics and Automation*, vol. 3, 01 2009.

[15] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proceedings of the 7th international joint conference on Artificial intelligencevol (IJCAI'81)*, vol. 2, pp. 674-679, 1981.

[16] K. Hata and S. Savarese, "Epipolar Geometry," *Stanford-CS231A*, p. 14, 2019.

[17] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "Icnet for real-time semantic segmentation on high-resolution images," *Computer Vision – ECCV 2018*, pp. 418-434, 2018.

[18] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016*

IEEE International Conference on Robotics and Automation (ICRA), pp. 1271–1278, 2016.

[19] A. Tourani, H. Bavle, J. L. Sanchez-Lopez, and H. Voos, "Visual slam: What are the current trends and what to expect?," *Sensors*, vol. 22, no. 23, 2022.

[20] J. Engel, T. Schöeps, and D. Cremers, "Lsd-slam: large-scale direct monocular slam," *European Conference on Computer Vision*, vol. 8690, pp. 1–16, 09 2014.

[21] K. Y. Kok and P. Rajendran, "A review on stereo vision algorithm: Challenges and solutions," *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, vol. 13, p. 112–128, Aug. 2019.

[22] K. E. Ozden, K. Schindler, and L. Van Gool, "Multibody structure-from-motion in practice," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 1134–1141, 2010.

[23] A. Kundu, K. M. Krishna, and C. V. Jawahar, "Realtime motion segmentation based multibody visual slam," ICVGIP '10, (New York, NY, USA), p. 251–258, Association for Computing Machinery, 2010.

[24] D. Zou and P. Tan, "Coslam: Collaborative visual slam in dynamic environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 354–366, 2013.

[25] S. Li and D. Lee, "Rgb-d slam in dynamic environments using static point weighting," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2263–2270, 2017.

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

[27] J. Cheng, Y. Sun, and M. Q.-H. Meng, "Improving monocular visual slam in dynamic environments: an optical-flow-based approach," *Advanced Robotics*, vol.33, no. 12, pp. 576–589,2019.

[28] R. Wang, W. Wan, Y. Wang, and K. Di, "A new rgb-d slam method with moving object detection for dynamic indoor scenes," *Remote Sensing*, vol. 11, no. 10, 2019.

[29] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, pp. 2980-2988, 2017.

[30] X. Long, W. Zhang, and B. Zhao, "Pspnet-slam: A semantic slam detect dynamic object by pyramid scene parsing network," *IEEE Access*, vol. 8, pp. 214685–214695, 2020.

[31] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Los Alamitos, CA, USA), pp. 6230–6239, IEEE Computer Society, jul 2017.

[32] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," 2015. arXiv:1511.00561. [Online]. Available: `https://doi.org/10.48550/arXiv.1511.00561`

[33] J. Cheng, Z. Wang, H. Zhou, L. Li, and J. Yao, "Dm-slam: A feature-based slam system for rigid dynamic scenes," *ISPRS International Journal of Geo-Information*, vol. 9, no. 4, 2020.

[34] L. Cui and C. Ma, "Sdf-slam: Semantic depth filter slam for dynamic environments," *IEEE Access*, vol. 8, pp. 95301–95311, 2020.

[35] C. Zhang, T. Huang, R. Zhang, and X. Yi, "Pld-slam: A new rgb-d slam method with point and line features for indoor dynamic scene," *ISPRS International Journal of Geo-Information*, vol. 10, p. 163, 03 2021.

[36] M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[37] Q.-T. Luong and O. Faugeras, "The fundamental matrix: Theory, algorithms, and stability analysis," *International Journal of Computer Vision*, vol. 17, pp. 43–75, 01 1996.

[38] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, p. 381–395, jun 1981.

[39] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, pp. 189-206, 2013.

[40] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 573–580, 2012.

Pattanapong Saeyong is currently pursuing the M. Eng. degree in Electrical Engineering, Prince of Songkla University, Hat Yai, Songkhla, Thailand. His current research interests include Visual SLAM, robotics, computer vision, and embedded systems.

Kittikhun Thongpull received the B.Eng. degree and the M.Eng. degree in Electrical Engineering, Prince of Songkla University, Hat Yai, Songkhla, Thailand, in 2008 and 2010, respectively. And the Ph.D. degree (Electrical and Computer Engineering) University of Kaiserslautern, Germany, in 2015.

Since September 2015, he has been a Faculty Member with the Department of Electrical Engineering, Faculty of Engineering, Prince of Songkla University. His research interests include wireless sensor networks, embedded systems, and industrial applications.