

# Towards Real-time Semantic RGB-D SLAM in Dynamic Environments

Tete Ji<sup>1</sup>, Chen Wang<sup>2</sup>, and Lihua Xie<sup>1</sup>

**Abstract**—Most of the existing visual SLAM methods heavily rely on a static world assumption and easily fail in dynamic environments. Some recent works eliminate the influence of dynamic objects by introducing deep learning-based semantic information to SLAM systems. However such methods suffer from high computational cost and cannot handle unknown objects. In this paper, we propose a real-time semantic RGB-D SLAM system for dynamic environments that is capable of detecting both known and unknown moving objects. To reduce the computational cost, we only perform semantic segmentation on keyframes to remove known dynamic objects, and maintain a static map for robust camera tracking. Furthermore, we propose an efficient geometry module to detect unknown moving objects by clustering the depth image into a few regions and identifying the dynamic regions via their reprojection errors. The proposed method is evaluated on public datasets and real-world conditions. To the best of our knowledge, it is one of the first semantic RGB-D SLAM systems that run in real-time on a low-power embedded platform and provide high localization accuracy in dynamic environments.

## I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) is a fundamental capability for intelligent robotic applications. It aims to simultaneously estimate the poses of the robot and build a map of the unknown environment from the data of on-board sensors [1], [2]. Benefiting from various low-cost and light-weight cameras available in the market, vision-based SLAM, or visual SLAM (vSLAM), has received increasing attention over the past decades. A number of impressive vSLAM systems have been proposed using different type of cameras, such as monocular SLAM [3], [4], RGB-D SLAM [5], [6], and stereo SLAM [7], [8]. Due to direct availability of the depth map and metric scale, RGB-D camera is a popular choice, especially for indoor scenes.

Although the existing vSLAM systems have achieved successful performance in some situations, the majority of these methods heavily rely on a static world assumption, which greatly limits their deployment in real world scenarios. Because dynamic objects such as moving people, animals and vehicles, have negative influence on the pose estimation and map reconstruction. Although robust estimation techniques such as RANSAC can be applied to filter out some of the outliers, the improvement is still limited since they can

This work was partly supported by the Delta-NTU Corporate Laboratory for Cyber-Physical Systems funded by the National Research Foundation (NRF) Singapore under its Corporate Laboratory@University Scheme.

<sup>1</sup>Tete Ji and Lihua Xie are with the School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798. {jite, elhxie}@ntu.edu.sg

<sup>2</sup>Chen Wang is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA. chenwang@dr.com

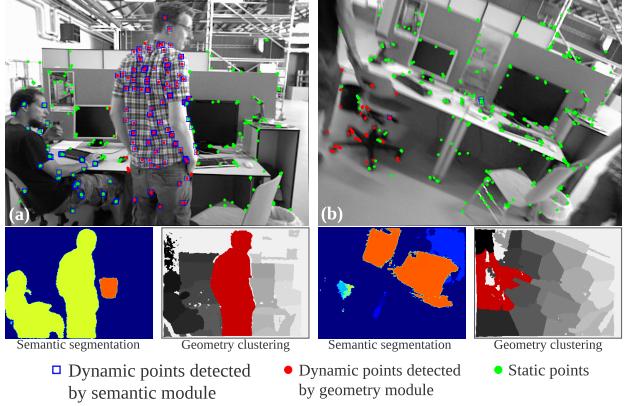


Fig. 1: An overview of dynamic objects detection by our proposed method. In case (a), the dynamic feature points associated with both the nearly static sitting person and the walking person are detected by the combination of semantic and geometry information. In case (b), the chair pulled by the person which is an unknown object to the semantic network is also successfully detected by our method.

only handle slightly dynamic scenes and may still fail when the moving objects cover most of the camera view.

Due to recent advances in computer vision and deep learning, semantic information of the environment has been integrated into SLAM systems, such as semantic mapping [9], [10] and object-level SLAM [11], [12]. The semantic information is extracted by semantic segmentation which predicts the labels and generates masks for detected objects. By recognizing and removing potentially dynamic objects, the performance of vSLAM in dynamic scenes can be greatly improved [13], [14]. However, there still exist two main problems for such methods. One is that most powerful deep neural networks for semantic segmentation such as Mask-RCNN [15] are highly computationally expensive and not applicable to real-time and small-scale robotic applications [16]. While for light-weight networks, the segmentation may be less precise and the tracking accuracy will also be affected. The other one is that they can only handle known objects which are labelled in the training set of the network, and may still fail when facing unknown moving objects.

To identify dynamic objects with semantic cues, most of the existing methods perform semantic segmentation on each new frame. This leads to a significant slowdown in camera tracking because the tracking process has to wait until the segmentation completes. Therefore, we extract semantic information only from keyframes to remove potentially

dynamic objects and maintain a map that contains only static features for camera tracking. Since keyframe and map update can run at low speed in a separate thread, the overall tracking time with semantic segmentation can be significantly reduced. Furthermore, to handle unknown moving objects, we propose an efficient geometry module that does not require prior information about the moving objects. This is done by segmenting the depth image into a few regions using K-Means algorithm and the dynamic regions are identified according to their average reprojection errors. Some examples are demonstrated in Fig. 1. Different from [17] which detects dynamic objects using geometry clustering in a dense optimization framework, we identify dynamic regions directly from reprojection errors of sparse features, which leads to faster processing and makes it more robust to dynamic contents.

The main contributions of this paper include:

- A keyframe-based semantic RGB-D SLAM system that is capable of reducing the influence of moving objects in dynamic environments.
- An effective and efficient geometry module that deals with unknown moving objects and combines with the semantic SLAM framework.
- Extensive evaluations showing that our method provides competitive accuracy compared to the state-of-the-art dynamic SLAM methods while being able to run in real-time on an embedded system.

## II. RELATED WORK

To deal with dynamic objects in the environments, a few vSLAM methods have been proposed in recent years. They can be divided into two main directions, one is geometry-based and the other is semantic or learning-based.

### A. Geometry-based Dynamic SLAM

The main idea of geometry-based methods for dealing with dynamic objects is to treat them as outliers and reject them using robust weighting functions or motion consistency constraints. Sun *et al.* [18] propose a motion removal approach that tracks and filters out motion patches in images using particle filter. Kim *et al.* [19] use a nonparametric model of the background and estimate the ego-motion only based on the estimated background model. StaticFusion [17] reconstructs the background by estimating a probabilistic segmentation of the image and integrating it in a weighted dense optimization framework. Zhang *et al.* [20] use dense optical flow residuals to detect dynamic regions in the scene and reconstruct the static background using similar framework to StaticFusion. More recently, Dai *et al.* [21] propose to use point correlations to distinguish static and dynamic map points based the fact that relative positions between static features are constant over time.

Geometry-based approaches do not need prior information about moving objects and the processing speed is generally faster. They are able to improve the performance of existing SLAM systems to some extent, but still need further efforts. Their accuracy is generally lower than that of semantic-based

methods. Moreover, geometry-based methods are unable to build semantic maps of the environment which is useful for advanced robotic applications.

### B. Semantic or Learning-based Dynamic SLAM

Learning-based methods make use of deep neural networks to extract semantic information in the environment by obtaining pixel-wise labels of input images and providing masks for each detected object. The potentially dynamic objects can then be identified from the semantic cues in a single image without the need of multi-frame processing. To get more consistent results, some learning-based methods also utilize conventional geometry information to further distinguish static and dynamic objects among the segmented objects. For example, DS-SLAM [22] combines SegNet and optical flow to identify moving people in the scene and removes dynamic features by checking the moving consistency. DynaSLAM [23] combines Mask-RCNN and multi-view geometry to handle both known and unknown moving objects. Although it achieved impressive improvement on public datasets, its computational cost is very high and is mainly for offline operation. In [24], a unified framework is introduced for dynamic SLAM and semantic segmentation which mutually improves the accuracy of each other.

Some methods continuously track the motion of the segmented dynamic objects instead of simply removing them. For example, MID-fusion [14] proposes an object-centric approach which incorporates object models in dense tracking formulation to track both object poses and camera motion. Differently, [25] proposes a novel motion model to track rigid moving objects which does not rely on their 3D models. The motion model and semantic information are then integrated in a factor graph optimization framework for objects and camera tracking. These methods achieved much improvement in dynamic environments, but all of them are either not able to run in real-time or not able to process unknown moving objects.

## III. PROPOSED METHOD

To deal with moving objects in dynamic environments, we make use of both semantic and geometry information from the input RGB-D images. Specifically, we take advantage of the learning-based methods for semantic information to process potentially dynamic objects, and we propose an efficient geometry module to deal with unknown moving objects. In order to achieve real-time performance, which is vital for robotic applications, we extract semantic cues only from keyframes with a light-weight network. Our system is built on ORB-SLAM2 [26], which is a feature-based SLAM system mainly for static environments. As our SLAM is sparse feature based, we only need to reject feature points associated with the moving objects. The overall framework is shown in Fig. 2, in which the semantic and the geometry module will be presented in the following sections.

### A. Semantic Module

The semantic segmentation is to predict pixel-wise labels and generate masks for detected objects in the input RGB

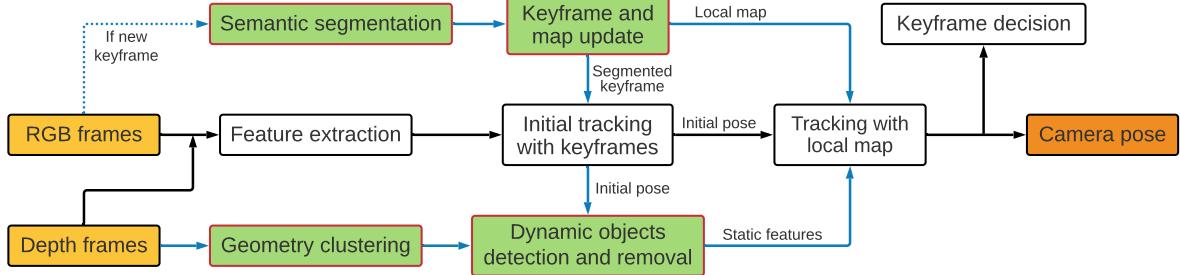


Fig. 2: The overall framework of proposed method. The black boxes are the tracking stages modified from ORB-SLAM2. The green-shaded boxes are our added stages to the SLAM system, which include a semantic module and a geometry module. The semantic module performs semantic segmentation on RGB image if it is a new keyframe, and then removes potentially dynamic objects in the keyframe and map. The geometry module performs geometry clustering on depth images and identifies moving objects based on the reprojection errors. After this, only static features are used for pose estimation.

images using deep learning-based methods. To account for both accuracy and computational speed, we adopted the light-weight semantic segmentation network SegNet [27] for our semantic module. Using more powerful deep neural networks like Mask-RCNN [15] can provide more precise segmentation results, however, the computation will be more expensive. The segmentation network is pre-trained on the PASCAL VOC dataset [28], which contains 20 classes of objects. Among these objects, we only deal with those that are highly movable or potentially dynamic, such as person, car, bicycle, etc. These objects will be removed from the segmented images and the feature points associated with them will not be used for camera tracking and map building. Instead of performing semantic segmentation for each new frame, which is standard in most existing learning-based dynamic SLAM methods, we do this only when a new keyframe is created. This significantly reduces the computational cost for the semantic module and helps us achieve real-time tracking with semantic information. Moreover, this process is performed in a separate thread, and hence does not have much influence on the overall tracking time.

To retain more information for tracking, some learning-based approaches further identify whether the segmented movable objects are moving or not. However for consistent long-term map building this is not necessary as these movable objects are undesired even when they are temporarily static. Hence we leave out this procedure in our semantic module. This shows another advantage of semantic information as it is able to recognize those undesired objects in the scene which are temporarily static and cannot be detected by conventional geometry-based approaches.

### B. Geometry Module

Since semantic information alone can only detect a fixed number of object classes that are labelled in the training set, the tracking and mapping will still be affected in the presence of unknown moving objects. Therefore, to handle unknown moving objects, we propose to combine a geometry module that does not require prior information about the objects.

To achieve this, we first segment each new depth image into  $N$  clusters using K-Means algorithm, where points close to each other in 3D space are grouped together. Each cluster is assumed to be a surface of an object and points within a cluster share the same motion constraint. Because a single object could be segmented into a few clusters, the object is not required to be rigid, while most semantic SLAM methods have this rigidity assumption. For each cluster  $c_j$ , we compute an average reprojection error  $r_j$  of all feature points  $\mathbf{u}_i$  inside the cluster against their matched correspondences  $\mathbf{P}_i$  in 3D space, as defined in (1), where  $m$  is the number of matched features in  $c_j$ ,  $\mathbf{T}_{wc}$  is the camera pose,  $\pi$  represents the camera projection model, and  $\rho$  is a penalty function.

$$r_j = \frac{1}{m} \sum_{i \in c_j}^m \rho \left( \|\mathbf{u}_i' - \pi(\mathbf{T}_{wc}\mathbf{P}_i)\|^2 \right) \quad (1)$$

When the error of a cluster is relatively larger than the others, we mark this cluster as dynamic. All feature points in dynamic clusters will be removed and not be involved in camera pose estimation. Comparing to identifying the dynamic state for each single feature point, our cluster-wise manner is more effective and efficient. Moreover, it prevents false detection caused by the measurement noise from a single point. It also allows us to approximate a rough shape of the moving objects via the geometry clustering. Some results of our method can be viewed in the third row of Fig. 3, where dynamic clusters are highlighted in red. This module can work independently without semantic information and hence can detect unknown moving objects.

The geometry module can also be regarded as a backup when semantic segmentation fails in detecting moving objects. This is because there exist situations that are challenging for semantic segmentation and may cause segmentation errors or failures, especially for light-weight networks. Some examples are presented in Fig. 3. It can be seen that although both modules can successfully detect moving objects in (a) and (b), the semantic segmentation fails in (c) to (g) due to large rotations, motion blurs, or only part of the

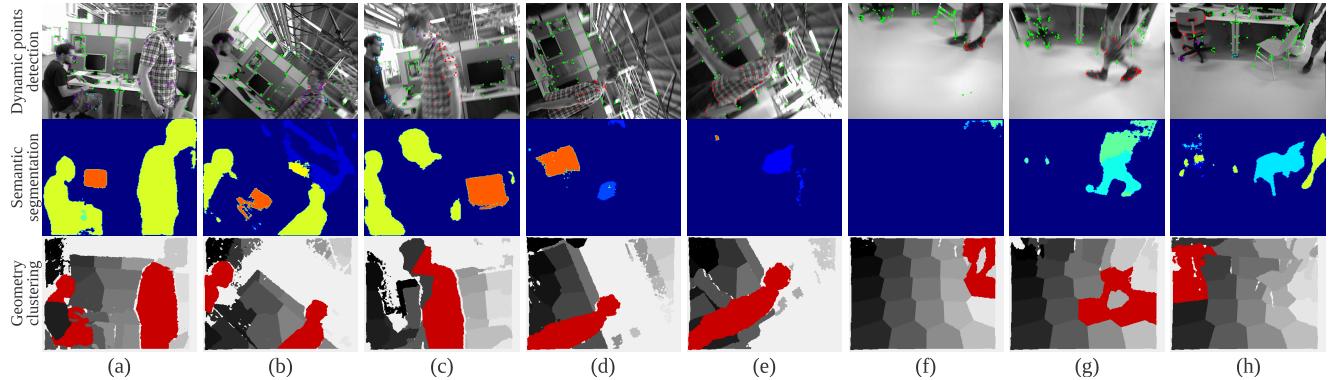


Fig. 3: Example results of dynamic objects detection. The first row shows dynamic features detected by the proposed semantic module (points with blue rectangle) and by the geometry module (points in red). The second row is the corresponding semantic segmentation result. The third row shows the geometry clustering results from depth images and the dynamic clusters are highlighted in red. (a) and (b) show that both modules detected dynamic objects. (c)-(h) demonstrate that semantic segmentation failed while geometry module succeeded. Frames are taken from the sequence *fr3/walking\_rpy*.

objects is present in the images. These situations introduce difficulties to semantic segmentation, while the geometry module can work as normal. Taking (c) as an example, the semantic segmentation network fails in recognizing a moving person due to severe motion blur caused by the fast relative motion of the person. While the features on this person are successfully detected by our geometry module because the reprojection errors of the dynamic parts are larger than the rest static regions. Note that we only plot tracked map points in the figure. The dynamic moving objects detected by the geometry module sometimes are not complete because there exist few matched map points in some regions of the objects. This is also a limitation of geometry methods which are difficult to detect a full boundary of moving objects.

An interesting thing we found during experiment is that some semi-dynamic objects could also be identified. An example is shown in Fig. 3 (h), where the left chair is identified to be dynamic. The reason is that the chair is currently static but its position was changed when revisiting it. This could be helpful for long-term consistent map building.

### C. Keyframe and Local Map Update

Most of the existing learning-based dynamic SLAM methods perform semantic segmentation for all new frames, which can only run at very low speed or offline mode. Different from that, we extract semantic information only from keyframes. Since a new frame is tracked with keyframes and local map, we only need to ensure that the segmented keyframes and the local map contain only static parts of the scene. The keyframe selection strategy is inherited from the original ORB-SLAM2 system and details will be omitted here. When a new keyframe is selected during tracking, we perform semantic segmentation in a separate thread and remove dynamic feature points. The local map is also updated by removing the corresponding dynamic map points. In this way, we maintain a keyframe database and a map that contain only static features and map points.

### D. Tracking

We perform a two-stage tracking for each new frame inspired by ORB-SLAM2. We first perform an initial tracking with a recent keyframe that has the largest overlap with the current frame to obtain an initial pose estimation. Since the keyframes have been refined with potentially dynamic objects removed, this initial estimation would be more reliable. The initial pose estimate is then used in the geometry module for dynamic objects detection. After removing dynamic points in the current frame by the geometry module, a more accurate pose estimation using local bundle adjustment is obtained by tracking with all the local map points observed in the current frame. As potentially dynamic map points are also removed in the local map by the semantic module, the influence of dynamic objects is further reduced and hence the pose estimation is more robust and accurate.

## IV. EXPERIMENTS AND RESULTS

*a) Overview:* The proposed method is tested on the TUM RGB-D dataset [29] which is widely used for the evaluation of RGB-D SLAM. Since our system is built on ORB-SLAM2, we use it as a baseline to demonstrate our improvement. We first evaluate the effect of different modules introduced in this paper. We then compare our results with the state-of-the-art dynamic SLAM methods including both geometry-based and learning-based approaches. The runtime analysis is also presented to show the efficiency of our method. Furthermore, we demonstrate the performance of our method with a live camera in real scenarios.

*b) Implementation:* For semantic segmentation, we use SegNet [27] with its default settings as described in Section III-A. For geometry module, we set the number of clusters  $N = 24$  at an image resolution of  $640 \times 480$ . Since false negatives (dynamic clusters marked as static) are unfavorable compared to false positives (static clusters marked as dynamic), we choose a relatively lower threshold for determining dynamic clusters, which depends on the average

TABLE I: Evaluation of ATE on the TUM dataset using the proposed method with different configurations [m]

Sequence	ORB-SLAM2	Geometry module	Semantic module	Combined system
fr3/s_xyz	<b>0.0094</b>	0.0100	0.0116	0.0117
fr3/s_half	0.0242	0.0187	0.0174	<b>0.0172</b>
fr3/w_static	0.1908	<b>0.0111</b>	0.0177	<b>0.0111</b>
fr3/w_rpy	0.8467	0.2143	0.0373	<b>0.0371</b>
fr3/w_xyz	0.4662	0.0290	0.0217	<b>0.0194</b>
fr3/w_half	0.4430	0.0362	0.0295	<b>0.0290</b>

reprojection error of matched features. All experiments are conducted on NVidia Jetson AGX Xavier with 8-core ARM v8.2 64-bit CPU, 512-core Volta GPU and 16G RAM, which is a low-power and small-size embedded platform designed for intelligent robotic applications. Our algorithm is tested on Ubuntu 18.04 with the Robot Operating System (ROS). To the best of our knowledge, our method is the first semantic RGB-D SLAM system for dynamic environments that is able to run in real-time on such an embedded platform.

c) *TUM Dataset*: The TUM RGB-D dataset contains 39 sequences captured in indoor environments with an RGB-D camera, and along with ground truth trajectories obtained from a high-accuracy motion capture system. To demonstrate the performance in dynamic environments, we selected a few sequences containing dynamic objects for evaluation. The *fr3/sitting* (*fr3/s* for short) sequences are slightly dynamic, where two persons sit at a desk with slight body motions. The *fr3/walking* (*fr3/w* for short) sequences are highly dynamic, where two persons walk around a desk. The *walking* sequences are most challenging since the dynamic objects occupy a large portion of the camera view. There are four types of camera motions as indicated in the sequence names, remaining almost static (*static*), moving along three directions (*xyz*), rotating along principal axes (*rpy*), and moving along a half sphere with 1-m diameter (*half*).

d) *Evaluation Metrics*: The error metrics for evaluation are the commonly-used Root-Mean-Square-Error (RMSE) of the Absolute Trajectory Error (ATE) in m, and RMSE of the Relative Pose Error (RPE) which comprises the translational drift in m/s and rotational drift in °/s. The ATE measures the the global consistency of the trajectory and the RPE measures the odometry drift per second.

#### A. The Effect of Different Modules

We first evaluate the accuracy of our system with different configurations including the geometry module, the semantic module, and their combined framework. The comparison of RMSE of the ATE against the baseline ORB-SLAM2 is shown in Table I. For slightly dynamic sequences, our proposed method provides similar results to ORB-SLAM2 since ORB-SLAM2 can handle these situations successfully by the RANSAC algorithm, and hence the improvement margin is limited. While for highly dynamic sequences, both of our semantic and geometry module achieve significant improvement in accuracy, and the proposed combined system

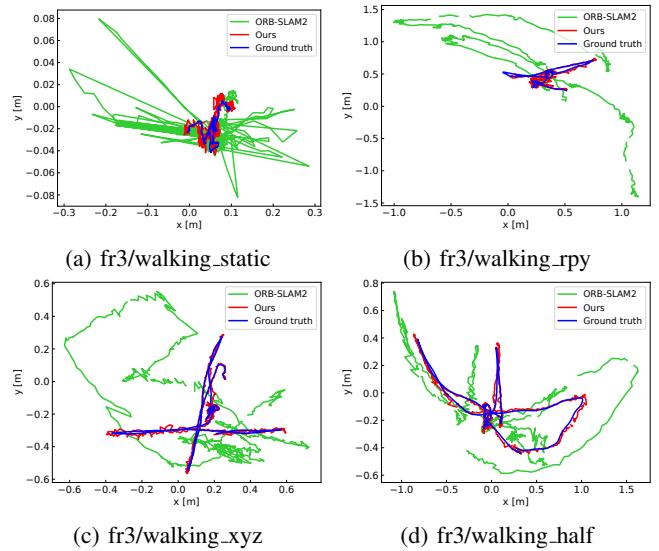


Fig. 4: Comparison of trajectories estimated by ORB-SLAM2 and the proposed method against ground truth.

achieves the best results. In fact, ORB-SLAM2 failed in tracking most of the time because of the dynamic objects. Therefore by removing the dynamic parts, we can obtain much more accurate camera pose estimations. This can also be seen in the trajectory estimation shown in Fig. 4.

#### B. Comparison with State-of-the-arts

We compare our results with state-of-the-art geometry-based dynamic SLAM methods, MR-DVO [18], SPW [30], StaticFusion [17], DSLAM [21], and learning-based methods, MID-Fusion [14], EM-Fusion [31], DS-SLAM [22], and DynaSLAM [23]. The comparisons of the ATE and RPE are summarized in Table II and Table III respectively. It can be seen that our method provides competitive results in all the dynamic sequences and outperforms all other dynamic SLAM methods except for DynaSLAM, which combines multi-view geometry in a semantic framework. However, it should be noted that DynaSLAM offers offline static map creation and is not able to run in real-time due to its time-consuming Mask-RCNN network and region growing algorithm. On the contrary, our method achieves real-time operation while providing very close results compared to it.

#### C. Runtime Analysis

To evaluate the efficiency of our proposed method, we measured the average computation time of each module and compared with the state-of-the-art learning-based methods DS-SLAM [22] and DynaSLAM [23], together with the baseline ORB-SLAM2. DS-SLAM and DynaSLAM are also built on ORB-SLAM2. The computation time is obtained using the sequence *fr3/walking\_xyz* on the same embedded platform. The results are listed in Table IV. It can be seen that our method is the only semantic RGB-D SLAM method that achieves real-time<sup>1</sup> on an embedded platform for dynamic

<sup>1</sup>Real-time in this paper means that a robot is able to process images as fast as a human brain, i.e., 100 ms per frame [32].

TABLE II: Comparison of RMSE of the Absolute Trajectory Error (ATE). The best results are highlighted in bold and the second-best are underlined. We use the results published in their original papers when applicable. [m]

Sequence	Geometry-based methods				Learning-based methods				Ours
	MR-DVO	SPW	StaticFusion	DSLAM	MID-Fusion	EM-Fusion	DS-SLAM	DynaSLAM	
fr3/sitting_xyz	0.0482	0.0397	0.040	<b>0.0091</b>	0.062	0.037	-	0.015	<u>0.0117</u>
fr3/sitting_half	0.0470	0.0432	0.040	<u>0.0235</u>	0.031	0.032	-	<b>0.017</b>	<b>0.0172</b>
fr3/walking_static	0.0656	0.0261	0.014	0.0108	0.023	0.014	<u>0.0081</u>	<b>0.006</b>	0.0111
fr3/walking_rpy	0.1333	0.1791	-	0.1608	-	-	0.4442	<b>0.035</b>	<u>0.0371</u>
fr3/walking_xyz	0.0932	0.0601	0.127	0.0874	0.068	0.066	0.0247	<b>0.015</b>	<u>0.0194</u>
fr3/walking_half	0.1252	0.0489	0.391	0.0354	0.038	0.051	0.0303	<b>0.025</b>	<u>0.0290</u>

TABLE III: Comparison of RMSE of the Relative Pose Error (RPE) in translational drift and rotational drift. \* indicates learning-based methods. Only a few learning-based methods reported RPE in their original papers.

Sequence	Translational RPE (m/s)				Ours	Rotational RPE (^/s)				Ours
	SPW	StaticFusion	DSLAM	DS-SLAM*		SPW	StaticFusion	DSLAM	DS-SLAM*	
fr3/sitting_xyz	0.0219	0.028	<b>0.0134</b>	-	0.0166	0.8466	0.92	<b>0.5792</b>	-	0.5968
fr3/sitting_half	0.0389	0.030	0.0354	-	<b>0.0259</b>	1.8836	2.11	0.8699	-	<b>0.7891</b>
fr3/walking_static	0.0327	0.013	0.0141	<b>0.0102</b>	0.0117	0.8085	0.38	2.7413	<b>0.2690</b>	0.2872
fr3/walking_rpy	0.2252	-	0.2299	0.1503	<b>0.0471</b>	5.6902	-	4.6327	3.0042	<b>1.0587</b>
fr3/walking_xyz	0.0651	0.121	0.1266	0.0333	<b>0.0234</b>	1.6442	2.66	2.7413	0.8266	<b>0.6368</b>
fr3/walking_half	0.0527	0.207	0.0517	<b>0.0297</b>	0.0423	2.4048	5.04	0.9854	<b>0.8142</b>	0.9650

TABLE IV: Comparison of Computation Time [ms]

Methods	Semantic part	Geometry part	Tracking
ORB-SLAM2	-	-	71.84
DS-SLAM	75.64	47.38	148.53
DynaSLAM	884.24	589.72	1144.93
Ours	72.36	30.14	75.82

environments. DS-SLAM uses the same network as ours and only deals with moving people, but it is still less efficient because it performs semantic segmentation for all frames. DynaSLAM has much higher processing time because its geometry part suffers from time-consuming region growing algorithm and sometimes it takes up to a few seconds to process if there are many dynamic points.

#### D. Robustness Test in Real Environment

We further tested our proposed method with a live RGB-D camera from Intel RealSense to evaluate its robustness in real dynamic environments. In our experiments, a person holding a book is sitting and walking in front of the camera and the camera is holding nearly static. Several screenshots of dynamic points detection results during the real-time test are shown in Fig. 5, where the second and third row are segmentation results from the semantic module and our proposed geometry module, respectively. Note that the book is not a labelled object in the network model and hence it cannot be recognized, or sometimes falsely recognized by the semantic module as shown in the second row. As a compensation process, the geometry module is able to correctly extract the book as a moving object in the test as shown in the third row. This demonstrates that both the semantic module and geometry module are necessary for robust semantic RGB-D SLAM system in dynamic environments. The average trajectory estimation error by our method is about 0.012m,

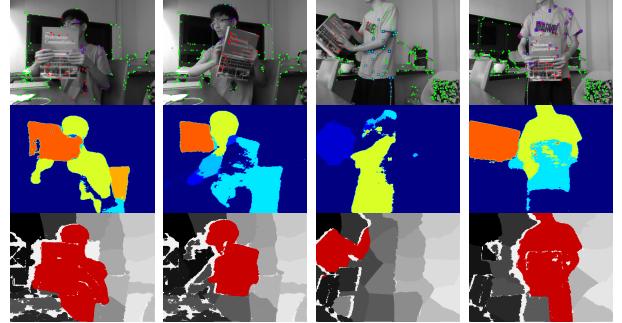


Fig. 5: Experiments with a live RGB-D camera in real dynamic scenarios. The dynamic features associated with both the known object (person) and unknown moving object (book) are successfully detected using our method.

while the error by ORB-SLAM2 is about 0.147m due to large fluctuations caused by the moving objects. This further demonstrates the effectiveness of our proposed method in the presence of both known and unknown moving objects.

## V. CONCLUSION

In this paper, we proposed a real-time semantic RGB-D SLAM framework for dynamic environments which is capable of processing both known and unknown moving objects. A keyframe-based semantic module is proposed in order to reduce the computational cost, and an effective geometry module using geometry clustering is introduced to deal with unknown moving objects. Extensive evaluations demonstrate that our system provides state-of-the-art localization accuracy while still being able to run in real-time on an embedded platform. In the future, we plan to build a long-term semantic map of the environment that contains only static parts, which is useful for high level robotic tasks.

## REFERENCES

- [1] S. Yuan, H. Wang, and L. Xie, "Survey on localization systems and algorithms for unmanned systems," *Unmanned Systems*, vol. 09, no. 02, pp. 129–163, 2021.
- [2] H. Wang, C. Wang, and L. Xie, "Intensity-slam: Intensity assisted localization and mapping for large scale environment," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1715–1721, 2021.
- [3] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European conference on computer vision*. Springer, 2014, pp. 834–849.
- [4] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [5] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 2100–2106.
- [6] C. Wang, J. Yuan, and L. Xie, "Non-iterative SLAM," in *2017 18th International Conference on Advanced Robotics (ICAR)*. IEEE, 2017, pp. 83–90.
- [7] J. Engel, J. Stückler, and D. Cremers, "Large-scale direct slam with stereo cameras," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 1935–1942.
- [8] R. Wang, M. Schworer, and D. Cremers, "Stereo dso: Large-scale direct sparse visual odometry with stereo cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3903–3911.
- [9] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks," in *2017 IEEE International Conference on Robotics and automation (ICRA)*. IEEE, 2017, pp. 4628–4635.
- [10] A. Hermans, G. Floros, and B. Leibe, "Dense 3d semantic mapping of indoor scenes from rgb-d images," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2631–2638.
- [11] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger, "Fusion++: Volumetric object-level slam," in *2018 international conference on 3D vision (3DV)*. IEEE, 2018, pp. 32–41.
- [12] S. Yang and S. Scherer, "Cubeslam: Monocular 3-d object slam," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 925–938, 2019.
- [13] M. Runz, M. Buffier, and L. Agapito, "Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects," in *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2018, pp. 10–20.
- [14] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger, "Mid-fusion: Octree-based object-level multi-instance dynamic slam," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5231–5237.
- [15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.
- [16] C. Wang, T. Ji, T.-M. Nguyen, and L. Xie, "Correlation flow: robust optical flow using kernel cross-correlators," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 836–841.
- [17] R. Scona, M. Jaimez, Y. R. Petillot, M. Fallon, and D. Cremers, "Staticfusion: Background reconstruction for dense rgb-d slam in dynamic environments," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–9.
- [18] 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.
- [19] D.-H. Kim and J.-H. Kim, "Effective background model-based rgb-d dense visual odometry in a dynamic environment," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1565–1573, 2016.
- [20] T. Zhang, H. Zhang, Y. Li, Y. Nakamura, and L. Zhang, "Flowfusion: Dynamic dense rgb-d slam based on optical flow," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 7322–7328.
- [21] W. Dai, Y. Zhang, P. Li, Z. Fang, and S. Scherer, "Rgb-d slam in dynamic environments using point correlations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
- [22] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "DS-SLAM: A semantic visual SLAM towards dynamic environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1168–1174.
- [23] 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.
- [24] K. Wang, Y. Lin, L. Wang, L. Han, M. Hua, X. Wang, S. Lian, and B. Huang, "A unified framework for mutual improvement of slam and semantic segmentation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5224–5230.
- [25] M. Henein, J. Zhang, R. Mahony, and V. Ila, "Dynamic slam: The need for speed," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2123–2129.
- [26] 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.
- [27] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.
- [28] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [29] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 573–580.
- [30] 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.
- [31] M. Strecke and J. Stuckler, "Em-fusion: Dynamic object-level slam with probabilistic data association," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5865–5874.
- [32] M. C. Potter and E. I. Levy, "Recognition memory for a rapid sequence of pictures." *Journal of experimental psychology*, vol. 81, no. 1, p. 10, 1969.