ASG-SLAM: An Adaptive Real-Time Visual Dynamic SLAM Based on Semantic Information and Geometric Constraints

Yun-Cong Ge[1,3,4], Chun-Yang Zhou[1,3,4], and Dan Chen[2,3,4],
and Zhen-Tao Liu[2,3,4,⋆]

[1] School of Future Technology, China University of Geosciences,
Wuhan 430074, China
[2] School of Automation, China University of Geosciences, Wuhan 430074, China
[3] Hubei Key Laboratory of Advanced Control and Intelligent Automation for
Complex Systems, Wuhan 430074, China
[4] Engineering Research Center of Intelligent Technology for Geo-Exploration,
Ministry of Education, Wuhan 430074, China

Abstract. Simultaneous Localization and Mapping (SLAM) is a core technology in the fields of autonomous driving and robotic navigation, and has made significant progress in recent years. However, conventional SLAM systems generally assume a static environment, which limits their practical application in dynamic real-world scenarios. To address the issues of pose estimation and map construction in dynamic environments by conventional SLAM, a novel real-time dynamic visual SLAM system (ASG-SLAM) is proposed in this paper. This system integrates advanced semantic segmentation models and geometric constraint methods to identify potential moving objects. It also utilizes prior semantic information to construct a motion probability grading model for objects, allowing the system to dynamically adjust its feature extraction strategy. These innovations provide ASG-SLAM with significant advantages in terms of localization accuracy, computational cost, and robustness. The system is evaluated on public datasets, and the results show that our method significantly improves both localization accuracy and computational efficiency compared to current state-of-the-art dynamic visual SLAM systems.

Keywords: Visual simultaneous localization and mapping (SLAM) · dynamic environment · instance segmentation · object detection

1 Introduction

Simultaneous Localization and Mapping (SLAM) has widespread applications in fields such as robotic navigation, augmented reality, and autonomous driving. SLAM technology enables robots or automatic systems to perform localization and map construction simultaneously in unknown environments using only data

[⋆] Corresponding author, E-mail: liuzhentao@cug.edu.cn streams from their own sensors. Visual SLAM, which primarily uses camera sensors, has received extensive attention from researchers in recent years due to the low power consumption, low cost, small size, and high integration of visual sensors. The rich information contained in images is not only suitable for the operation of SLAM systems themselves but also supports vision-based applications such as semantic segmentation and object detection. Advanced visual SLAM systems, such as ORB-SLAM2 [1] and LSD-SLAM [2], and others [3], [4]. have achieved impressive performance.

The application of visual SLAM algorithms in reality is often limited by their fundamental assumption of static environments. When the system operates in scenes containing a large number of dynamic objects, conventional methods can fail due to feature points on dynamic objects, leading to feature association failures. This not only significantly increases the error in pose estimation but may also affect the stability of the system. For example, ORB-SLAM2 [1] often encounters failures when dealing with the highly dynamic TUM RGB-D DynamicObjects dataset [5], leading to operational failures. This challenge indicates that the design of visual SLAM algorithms needs to take dynamic environmental factors into more consideration to ensure robustness and practicality in real-world scenarios such as service robots [6], autonomous driving, and augmented/virtual reality (AR/VR).

In recent years, many researchers have developed various algorithms to enhance the performance of SLAM systems in dynamic environments. These algorithms primarily focus on the detection and exclusion of dynamic objects. DynaSLAM [7] utilizes multi-view geometry techniques [9] to assist the Mask R-CNN [8] instance segmentation network, achieving more precise image segmentation and effectively detecting and repairing the background around moving objects in images. DS-SLAM [10] combines the SegNet [11] semantic segmentation network with optical flow technology to remove dynamic feature points from images. Additionally, YOLO-SLAM [12] integrates the YOLOv3 [13] object detection network to preliminarily identify dynamic objects in the scene and excludes feature points within these objects' range by combining with the RANSAC [14] method. SG-SLAM [15] further identifies dynamic objects by combining the SSDlite [16] object detection network with epipolar constraint technology.

To address the challenges posed by dynamic objects to SLAM algorithms, this paper introduces a visual SLAM method designed specifically for the removal of dynamic objects (ASG-SLAM). ASG-SLAM utilizes semantic information and geometric constraint methods to identify potential moving objects. It also employs prior semantic information to establish a motion probability grading model, allowing the system to adaptively adjust its feature extraction strategy. These methods alleviate the impact of dynamic objects within the visual SLAM system, thereby enhancing the system's localization accuracy and robustness in complex dynamic environments to meet the operational needs of robots in complex scenarios. Figure 1 shows an overview of the ASG-SLAM system.

The main contributions of this paper are summarized as follows:

- In this paper, we have integrated real-time object detection and image segmentation technologies using YOLOv8 into the ORB-SLAM2 [1] framework, introducing a parallel instance segmentation thread to generate basic semantic information of potential dynamic objects in the scene in real-time. Additionally, geometric constraint methods have been incorporated into the existing feature extraction thread to improve accuracy. This system is capable of achieving high precision and robustness in real-time visual SLAM in dynamic environments.
- A motion probability grading model is proposed in this paper, which is drawn on real-world experience by assigning different prior motion probabilities to objects of different semantic types. When potentially moving objects constitute a significant proportion of the environment, the system adjusts its feature extraction strategy, specifically removing feature points from objects with high motion probabilities. This method significantly enhances the robustness and accuracy of the system in dynamic scenes.
- We evaluated the performance of ASG-SLAM on the TUM RGB-D DynamicObjects dataset [5]. The comparison results demonstrate that ASG-SLAM significantly outperforms ORB-SLAM2 [1] in high dynamic environments, both in terms of pose estimation accuracy and system robustness. Its performance also surpasses other leading visual dynamic SLAM systems.

The structure of the remaining sections of this paper is as follows. Section 2 elaborates on the framework of the entire SLAM system, explaining how dynamic objects are detected and an adaptive feature extraction strategy is implemented. Section 3 rigorously tests the performance of ASG-SLAM, demonstrating the effectiveness and accuracy of the system. Section 4 concludes with a brief summary and discusses future research plans.

2 Methodology

This section provides a detailed introduction to the framework of the ASG-SLAM system, covering four aspects. First, we present an overview diagram of the ASG-SLAM system. Second, we briefly introduce the design of the real-time instance segmentation network thread based on YOLOv8 within ASG-SLAM. Next, we describe a dynamic point exclusion method based on motion consistency detection. Finally, we demonstrate how to construct a motion probability grading model and design a strategy for the exclusion of dynamic feature points.

2.1 Framework of ASG-SLAM

The ASG-SLAM system is based on the existing ORB-SLAM2 [1] framework, seamlessly integrating the high-performance SLAM capabilities of ORB-SLAM2 [1] with a real-time instance segmentation thread and dynamic feature point exclusion thread based on YOLOv8. This system helps robots maintain good pose estimation performance in dynamic environments and complete the construction

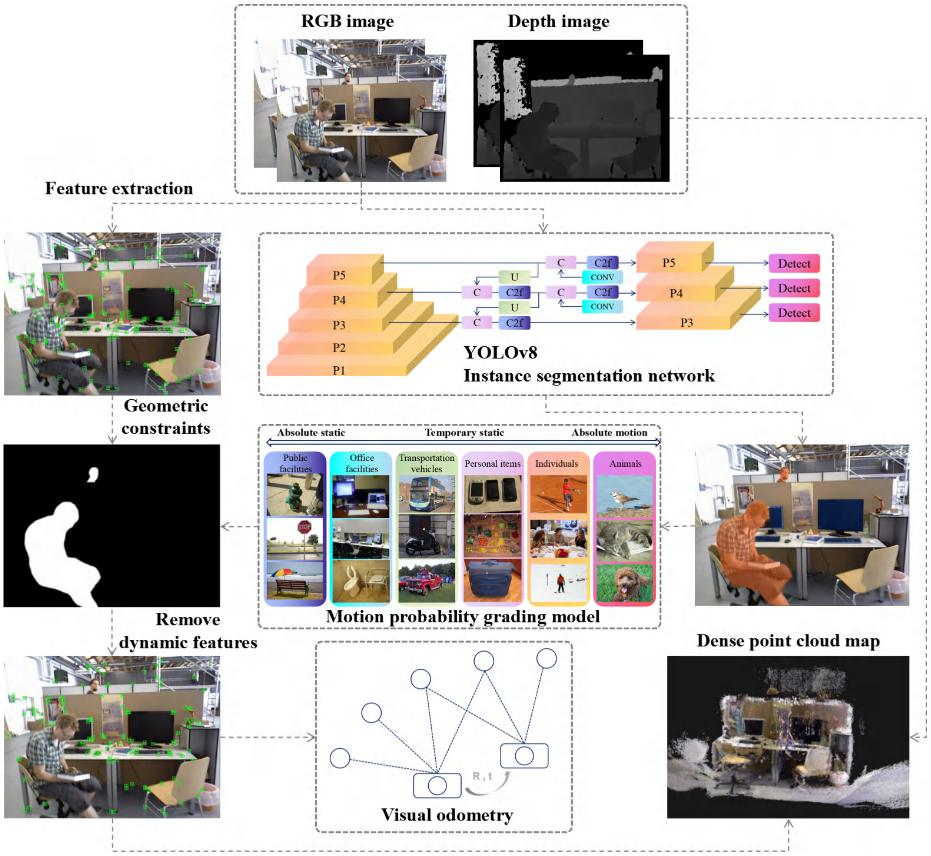


Fig. 1: Overview of the ASG-SLAM framework: the original RGB images simultaneously enter the feature extraction and instance segmentation threads. After excluding dynamic features, pose estimation is performed, and point cloud maps are constructed using depth images.

of dense point cloud maps. During operation, two main threads run in parallel: the feature extraction thread and the instance segmentation thread. The overall framework is illustrated in Figure 1.

The feature extraction thread takes RGB images as input, responsible for estimating the camera pose and deciding whether to insert keyframes. It employs advanced methods to cope with dynamic environments, including ORB feature extraction and matching, capturing feature point movements between consecutive frames using optical flow pyramid techniques, obtaining matched feature point pairs between previous and subsequent frames, and estimating the current fundamental matrix using the random sample consensus method. It then uses epipolar constraints to identify and exclude dynamic feature points.

Simultaneously, the real-time instance segmentation thread based on YOLOv8 performs instance segmentation on RGB images and obtains semantic information. By the judgment of a motion probability model and considering the available number of static feature points, the most likely moving objects in the image are identified and masks are created. These masks then collaborate with geometric constraint methods to minimize dynamic features to the greatest extent possible. Both parallel threads work simultaneously to construct a semantically rich map.

2.2 Instance Segmentation Based on YOLOv8

In the fields of object detection and instance segmentation, the YOLOv8 model has significantly enhanced efficiency and accuracy through its innovative architectural design. This model adopts a one-stop prediction strategy, dividing the image into multiple grids and predicting the category, location, and segmentation mask of objects within each grid. This approach optimizes processing speed and enhances prediction accuracy. The architecture of YOLOv8 is divided into two parts: the Backbone network based on Darknet and the Head section.

The Darknet network is a lightweight structure designed for speed optimization, effectively extracting image features through a combination of convolutional and pooling layers. YOLOv8 further incorporates residual connections and skip connections, which not only facilitate deep information transmission but also enhance the detection capabilities for small-sized targets. The Head section introduces attention mechanisms and multi-scale feature fusion. The attention mechanism adaptively adjusts the distribution of feature map weights, enhancing the model's focus on key information and thereby improving prediction accuracy. Multi-scale feature fusion enhances the model's perception of targets of various sizes by combining features from different levels, which is crucial for instance segmentation in complex scenes.

During the training phase, YOLOv8 adopts a stepwise strategy: initially performing object detection to generate candidate boxes, then using these boxes to produce and optimize segmentation masks during the instance segmentation stage. This method not only improves training efficiency but also optimizes the model's performance in practical applications. YOLOv8 has broad application prospects in fields such as image processing and machine learning.

2.3 Motion Consistency Detection

Regarding the geometric constraint methods, we have employed a detection method based on motion consistency to identify dynamic feature points in images. The procedure is performed as follows: First, we utilize optical flow pyramid technology to capture the movement of feature points between consecutive frames by analyzing images across multiple scales. This allows us to acquire matched feature pairs between the current and previous frames. We discard pairs located within potential moving objects or at the image edges. Subsequently, we process the remaining matched pairs using the Random Sample Consensus

(RANSAC) [14] method to estimate the current fundamental matrix. Using this fundamental matrix, we calculate the epipolar lines for the matched pairs in the current frame. Lastly, we evaluate whether the distance from the matched points to the epipolar lines in the current frame falls below a predefined threshold. If it is below the threshold, the point is considered stationary; otherwise, it is deemed to be moving.

Figure 2 provides a diagram of epipolar geometric constraints, where P represents a point in three-dimensional space, O_1 and O_2 represent the camera centers of two frames, l_1 and l_2 represent the epipolar lines, and p_1 and p_2 represent the matched points in the previous and current frames, respectively. According to [9], The fundamental matrix F can be calculated by

$$p_2^T F p_1 = 0 \tag{1}$$

The epipolar line l_2 can be calculated using

$$l_2 = FP_2 = F \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} \tag{2}$$

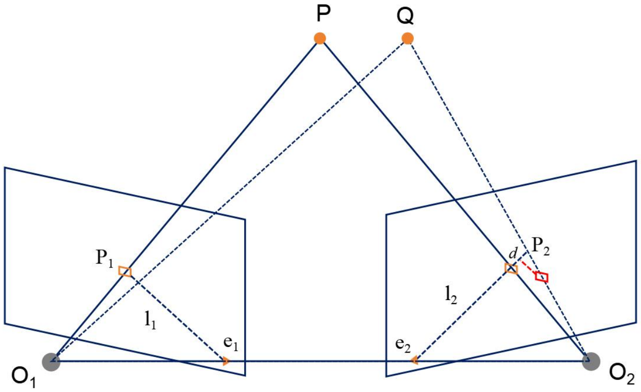


Fig. 2: Epipolar Constraints.

2.4 Motion Probability Grading Model

Relying solely on semantic information extracted from images by instance segmentation networks to determine the static or dynamic nature of objects may not always be reliable. For example, a person standing still is stationary, but becomes dynamic when they start moving. To address this challenge, this paper proposes a motion probability grading model. This model assigns different prior motion probabilities to objects of various semantic categories based on real experience, rather than simply classifying them as static or dynamic. Based on everyday observations, we categorize the labels provided by the COCO dataset [17] into six categories, each sorted by the likelihood of motion.

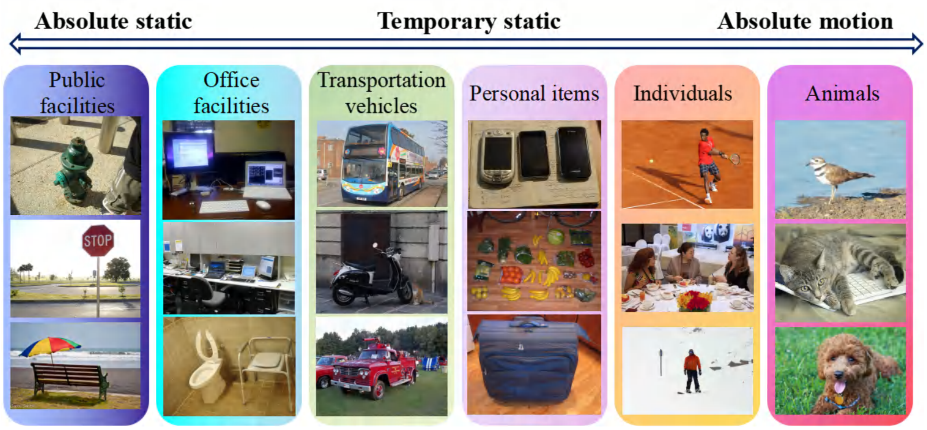


Fig. 3: Probabilistic motion grading model: Different objects are associated with different motion probabilities. This model balances maintaining a sufficient number of feature points and removing dynamic features.

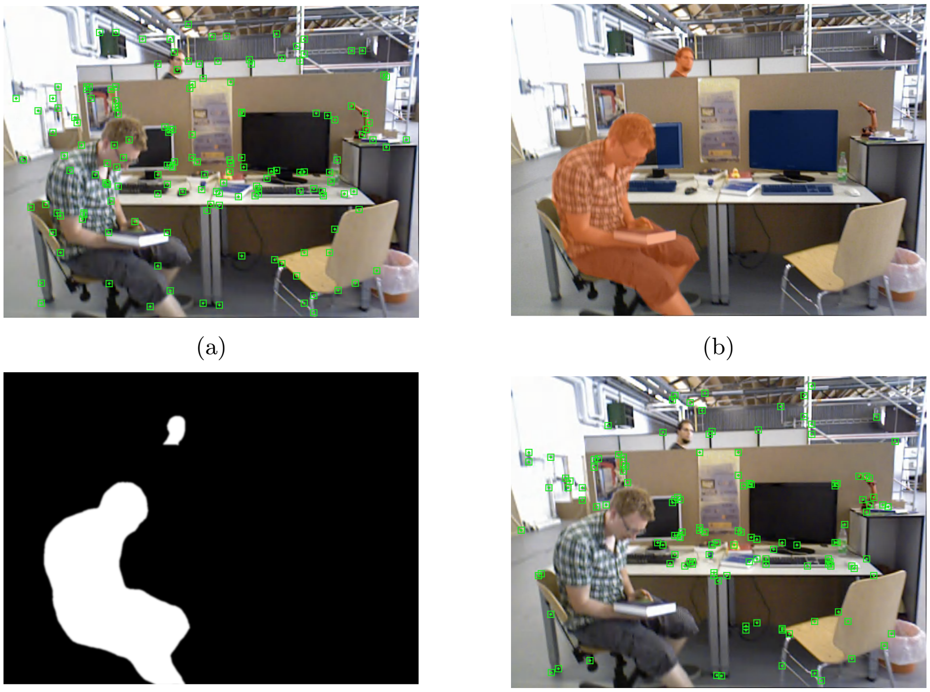
As shown in Figure 3, we have categorized objects as follows: The first category includes sturdy public facilities; the second category comprises common household and office items; the third category involves various types of transportation vehicles, such as buses and motorcycles; the fourth category consists of personal belongings; the fifth category includes humans; and the sixth category encompasses animals such as birds and cats. The feature points in the image are assigned prior motion probabilities based on the semantic classification masks to which they belong.

In conjunction with the motion probability model, we have designed a method aimed at addressing the issue of system tracking failure due to a lack of feature points when there are many feature points on potentially moving objects in the image. Initially, we define the number of static feature points in the current frame as m and the total number of feature points as n, then the ratio δ of the two can be calculated using

$$\delta = \frac{m}{n} \tag{3}$$

We can set a threshold δ_1 based on the actual conditions of the environment in which the system operates. If the proportion of static feature points available in the current frame is below the threshold δ_1, feature points that were excluded from potentially moving objects are progressively restored in order of increasing motion probability until the proportion of static feature points reaches or exceeds δ_1. Thanks to the efficient parallel thread design, this method can be implemented in a very short time after instance segmentation is completed. By setting an appropriate δ_1, a balance can be found between maintaining a sufficient number of feature points and removing dynamic feature points, significantly enhancing the system's robustness.

3 Experiments and Analysis



(a)

(b)

(c)

(d)

Fig. 4: Motion object feature point rejection flowchart:(a) Distribution of ORB feature points extracted by the tracking thread; (b) Semantic mask image obtained from the instance segmentation network; (c) Final mask image derived from the instance segmentation network, geometric constraints, and the probabilistic motion model; (d) Distribution of ORB feature points after dynamic features are excluded by the ASG-SLAM system.

In this section, we evaluate the performance of ASG-SLAM in dynamic environments using the publicly available TUM RGB-D dataset [5] from the Technical University of Munich. This dataset includes multiple dynamic sequences in which the scenes contain a rich array of potentially moving objects, with these objects occupying more than half of the image area, presenting significant challenges for visual SLAM systems. This experiment conducts a rigorous test of the accuracy and robustness of our proposed SLAM system.

All the experiments in this study were conducted on a local laptop equipped with an AMD Ryzen 7 5800H CPU, 16GB RAM, and a GeForce RTX 3060 graphics card with 6GB of VRAM.

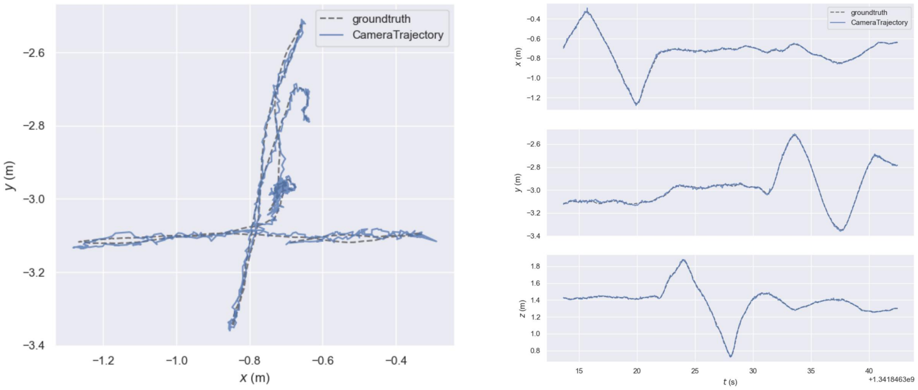
3.1 Experiment on Eliminating Potentially Dynamic Objects

To evaluate the effectiveness of ASG-SLAM in excluding dynamic feature points, we conducted Experiment 1. Figure 4 (a) illustrates the scenario where two pedestrians are considered potential dynamic objects, showing the numerous feature points extracted by the conventional ORB-SLAM2 [1] system on these individuals. Figure (b) presents the semantic mask image obtained by applying the YOLOv8 instance segmentation network. Objects with a high activity probability, such as the seated and standing individuals, are marked with an orange-red mask, while objects with a low activity probability, such as the computer and keyboard, are marked with a blue mask. Figure (c) shows the integration of instance segmentation, geometric constraints, and the motion probability model by the system. The motion probability model recovered potential dynamic feature points on the computer and keyboard, identified the two high-motion probability pedestrians, and generated the final mask image. Figure (d) displays the distribution of ORB feature points in the ASG-SLAM system after removing the dynamic feature points, clearly showing the removal of feature points from the high-activity probability individuals while retaining the feature points on the low-motion probability computer and keyboard, supporting subsequent pose estimation and map construction.

3.2 Performance Evaluation in Dynamic Environments

This study conducted Experiment 2 on the TUM RGB-D dataset [5], which involved the following dataset sequences: *fr3_walking_xyz*, *fr3_walking_static*, *fr3_walking_rpy*, *fr3_walking_half*, and *fr3_sitting_static*. These sequences captured scenes of walking and sitting. The dynamics of the scenes are indicated by the labels *xyz*, *static*, *rpy*, and *half*, representing different camera movement patterns along various axes.

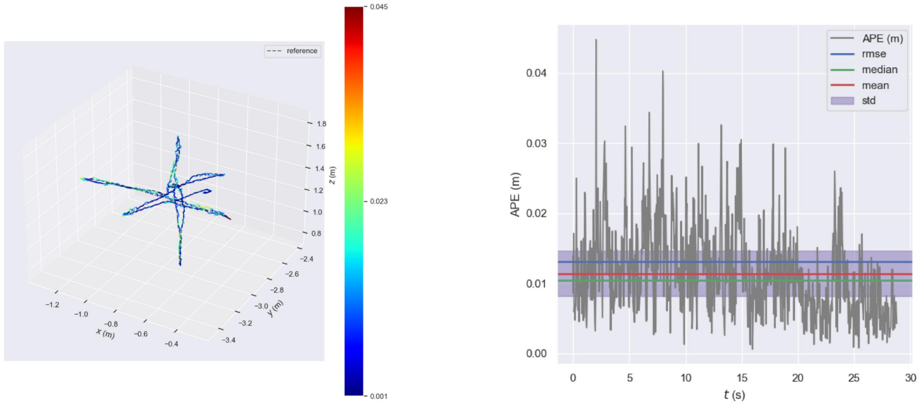
To quantitatively evaluate the performance of the algorithm, we employed the Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) metrics [5]. ATE assesses the global consistency of the trajectory, while RPE focuses on the translational and rotational drift of the algorithm over time.We conducted comprehensive tests on the performance of ASG-SLAM using the *fr3_walking_xyz* sequence. Figure 5 (a) presents the trajectory of the SLAM system in the *xy* plane, while Figure 5 (b) shows the trajectory performance along the *xyz* coordinate axes. Figure 6 (a) illustrates the ATE plot, where the gray dashed line represents the ground truth reference trajectory, and the colored solid lines depict the estimated poses by the SLAM system. Figure 6 (b) displays the ATE



(a) (b)

Fig. 5: (a)The trajectory plot in the xy plane.(b)The trajectory plot along the xyz coordinate axes.

evaluation curve. The results indicate that ASG-SLAM has a low trajectory error and did not experience tracking loss during the testing process.



(a) (b)

Fig. 6: (a)The ATE trajectory plot.(b)The ATE plot.

To further evaluate the performance of ASG-SLAM, we tested five visual SLAM systems on the selected sequences. These systems include ORB-SLAM2 [1], which excels in static environments, and the state-of-the-art dynamic SLAM solutions Dyna-SLAM [7], DS-SLAM [10], and YOLO-SLAM [12], all of which are built on the ORB-SLAM2 [1] framework. We calculated the Root Mean Square Error (RMSE) and Standard Deviation (S.D.) of the Absolute Trajectory Error

(ATE) and Relative Pose Error (RPE) [5]for all dataset sequences. The results are presented in Tables 1, 2, and 3. The best performance for each evaluation metric is highlighted in bold.

From the data in Tables 1 to 3, it is evident that ASG-SLAM significantly outperforms ORB-SLAM2 [1] in terms of Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) [5], achieving an overall improvement in localization performance by an order of magnitude. Compared to DynaSLAM [7], DS-SLAM [10], and YOLO-SLAM [12], ASG-SLAM only slightly underperforms DynaSLAM in the ATE for the *fr3_walking_static* sequence, while maintaining a leading position in all other dataset sequences.

Table 1: Analysis of absolute trajectory error (ATE[m]) metric.

Sequences	ORB-SLAM2		DynaSLAM		DS-SLAM		YOLO-SLAM		**ASG-SLAM**	
	RMSE	*S.D.*	*RMSE*	*S.D.*	*RMSE*	*S.D.*	*RMSE*	*S.D.*	*RMSE*	*S.D.*
fr3_w_x	0.7521	0.3759	0.0164	0.0086	0.0247	0.0161	0.0146	0.0070	**0.0130**	**0.0063**
fr3_w_s	0.3900	0.1602	**0.0068**	**0.0032**	0.0081	0.0036	0.0073	0.0035	0.0071	0.0034
fr3_w_r	0.8705	0.4520	0.0354	0.0190	0.4442	0.2350	0.2164	0.1001	**0.0325**	**0.0171**
fr3_w_h	0.4863	0.2290	0.0296	0.0157	0.0303	0.0159	0.0283	0.0138	**0.0256**	**0.0134**
fr3_s_s	0.0087	0.0043	0.0108	0.0056	0.0065	0.0033	0.0066	0.0033	**0.0062**	**0.0031**

Table 2: Analysis of translational relative pose error (RPE[m]) metric.

Sequences	ORB-SLAM2		DynaSLAM		DS-SLAM		YOLO-SLAM		**ASG-SLAM**	
	RMSE	*S.D.*	*RMSE*	*S.D.*	*RMSE*	*S.D.*	*RMSE*	*S.D.*	*RMSE*	*S.D.*
fr3_w_x	0.4124	0.2684	0.0217	0.0119	0.0333	0.0229	0.0194	0.0097	**0.0117**	**0.0068**
fr3_w_s	0.2162	0.1962	0.0089	0.0044	0.0102	0.0048	0.0094	0.0044	**0.0071**	**0.0040**
fr3_w_r	0.4249	0.3166	0.0448	0.0262	0.1503	0.1168	0.0933	0.0736	**0.0346**	**0.0174**
fr3_w_h	0.3550	0.2810	0.0284	0.0149	0.0297	0.0152	0.0268	0.0124	**0.0199**	**0.0102**
fr3_s_s	0.0095	0.0046	0.0126	0.0067	0.0078	0.0038	0.0089	0.0044	**0.0052**	**0.0026**

Table 3: Analysis of rotational relative pose error (RPE[deg]) metric.

Sequences	ORB-SLAM2		DynaSLAM		DS-SLAM		YOLO-SLAM		**ASG-SLAM**	
	RMSE	*S.D.*	*RMSE*	*S.D.*	*RMSE*	*S.D.*	*RMSE*	*S.D.*	*RMSE*	*S.D.*
fr3_w_x	7.7432	4.9895	0.6284	0.3848	0.8266	0.5826	0.5984	0.3655	**0.4338**	**0.2228**
fr3_w_s	3.8952	3.5095	0.2612	0.1259	0.2690	0.1182	0.2623	0.1104	**0.2080**	**0.0964**
fr3_w_r	8.0802	5.9499	0.9894	0.5701	3.0042	2.3065	1.8238	1.4611	**0.6077**	**0.2826**
fr3_w_h	7.3744	5.7558	0.7842	0.4012	0.8142	0.4101	0.7534	0.3564	**0.5806**	**0.3136**
fr3_s_s	0.2881	0.1244	0.3416	0.1642	0.2735	0.1215	0.2709	0.1209	**0.1980**	**0.0884**

Table 4 presents the inference time results of the deep learning network models and the hardware platforms used in the experiments. In these experiments, ASG-SLAM's instance segmentation network demonstrates excellent performance, with an average inference speed of 23 milliseconds per image, significantly faster than the segmentation speeds of other SLAM systems designed for dynamic environments.

Table 4: Time analysis

Systems	Neural Network	Average Inference Time Per Frame (ms)	Hardware Platform
DS-SLAM	SegNet	37.57	Intel i7, P4000
Dyna-SLAM	Mask R-CNN	195	Nvidia Tesla M40
ASG-SLAM	YOLOv8-seg	22.88	AMD R7 5800H,RTX 3060

The research results indicate that in highly dynamic environments, ASG-SLAM is capable of maintaining low levels of Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) [5], significantly outperforming conventional visual SLAM algorithms such as ORB-SLAM2 [1]. Additionally, compared to most advanced SLAM systems designed for dynamic scenes, such as DynaSLAM [7], DS-SLAM [10], and YOLO-SLAM [12], ASG-SLAM not only demonstrates superior performance but also maintains higher efficiency.

4 Conclusion

A novel SLAM system named ASG-SLAM is introduced in this paper, designed to minimize the interference of dynamic objects on localization.Based on ORB-SLAM2 [1], ASG-SLAM features parallel instance segmentation and tracking threads. We integrate an instance segmentation network based on YOLOv8 and geometric constraints to acquire rich semantic information from the environment and effectively eliminate feature points on potential dynamic objects. Additionally, we have developed a motion probability grading model, allowing the system to balance between maintaining a sufficient number of feature points and removing dynamic ones. These innovative designs enhance the robustness and accuracy of the system in dynamic scenes. Experimental results demonstrate that ASG-SLAM significantly surpasses conventional methods in accuracy and robustness within complex dynamic environments and exhibits excellent performance in image inference speed of the segmentation network. Despite these advancements, ASG-SLAM still requires improvements. In the future, we plan to introduce advanced techniques such as reinforcement learning to aid the system's localization and mapping, enhancing the autonomous navigation capabilities of robots in unknown and extreme environments.

References

1. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. IEEE Trans. Robot. **33**(5), 1255–1262

(2017)

2. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: Large-scale direct monocular SLAM. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8690, pp. 834–849. Springer, Cham (2014)
3. Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M.M., Tardós, J.D.: ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM. IEEE Trans. Robot. **37**(6), 1874–1890 (2021)
4. Qin, T., Li, P., Shen, S.: VINS-mono: A robust and versatile monocular visual-inertial state estimator. IEEE Trans. Robot. **34**(4), 1004–1020 (2018)
5. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., pp. 573–580 (2012)
6. Chen, D., Ge, Y.: Multi-objective navigation strategy for guide robot based on machine emotion. Electronics **11**(16), 2482 (2022)
7. Bescos, B., Fácil, J.M., Civera, J., Neira, J.: DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes. IEEE Robot. Autom. Lett. **3**(4), 4076–4083 (2018)
8. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proc. IEEE Int. Conf. Comput. Vis. (ICCV), pp. 2980–2988 (2017)
9. Kundu, A., Krishna, K.M., Sivaswamy, J.: Moving object detection by multi-view geometric techniques from a single camera mounted robot. In: Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., pp. 4306–4312 (2009)
10. Yu, C., Liu, Z., Liu, X.-J., Xie, F., Yang, Y., Wei, Q., Fei, Q.: DS-SLAM: A semantic visual SLAM towards dynamic environments. In: Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), pp. 1168–1174 (2018)
11. Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: A deep convolutional encoder–decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(12), 2481–2495 (2017)
12. Wu, W., Guo, L., Gao, H., You, Z., Liu, Y., Chen, Z.: YOLO-SLAM: A semantic SLAM system towards dynamic environment with geometric constraint. Neural Comput. Appl. **34**(8), 6011–6026 (2022)
13. Redmon, J., Farhadi, A.: YOLOv3: An incremental improvement. arXiv:1804.02767 (2018)
14. Fischler, M.A., Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
15. Cheng, S., Sun, C., Zhang, S., Zhang, D.: SG-SLAM: A real-time RGB-D visual SLAM toward dynamic scenes with semantic and geometric information. IEEE Trans. Instrum. Meas. **72**, 1–12 (2023)
16. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: SSD: Single shot MultiBox detector. In: Computer Vision—ECCV 2016, pp. 21–37. Springer, Amsterdam (2016). https://doi.org/10.1007/978-3-319-46448-0_2
17. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: Computer Vision–ECCV 2014, pp. 740–755. Springer, Zurich (2014). https://doi.org/10.1007/978-3-319-10602-1_48