Available Online: 26 March 2026 DOI: 10.54254/2977-3903/2026.32445

Visual SLAM algorithm with dynamic point elimination based on YOLACT network

Jiahui Zhou

School of Artificial Intelligence and Computer Science, North China University of Technology, Beijing, China

zhou423_jiahui@163.com

Abstract. A dynamic visual Simultaneous Localization and Mapping (SLAM) algorithm is proposed in this paper, which combines the YOLACT network with the geometric method to design a dynamic point detection module for eliminating dynamic points. The dense optical flow-based dynamic point detection scheme is adopted to make up for the problem that the elimination algorithm based on the instance segmentation network overrelies on object prior information. Aiming at the low accuracy of the original output mask of YOLACT, a mask post-processing method based on image processing and morphology is proposed to repair the dynamic point mask output by the YOLACT network. Finally, this module is integrated into the Oriented FAST and Rotated BRIEF SLAM 2 (ORB-SLAM2) framework to construct a visual SLAM system adapted to dynamic scenes. The proposed algorithm is tested and verified on the public TUM dataset, which proves the effectiveness of the proposed module. Compared with the ORB-SLAM2 system, the localization accuracy of the proposed algorithm is improved by 93.4% in indoor dynamic scenes.

Keywords: visual SLAM, dynamic scenes, instance segmentation, optical flow method

1. Introduction

Simultaneous Localization and Mapping (SLAM) technology is the core technology in the fields of mobile robots, autonomous driving and augmented reality. It endows mobile robots with the ability of ego pose estimation and construction of metric scene maps in unknown environments, and is the core cornerstone for achieving high autonomy [1]. In recent years, classic visual SLAM systems have achieved milestone success with their excellent accuracy and generalization ability, but their underlying multi-view geometric frameworks all highly rely on the strict static environment assumption [2]. However, pedestrians, moving vehicles or moved objects are inevitably present in real complex scenes. The feature points generated by these dynamic objects will be mistaken for the camera ego-motion by the algorithm, leading to serious data association errors, and ultimately causing huge drift in camera pose estimation or even system crash. Therefore, how to enhance the robustness of visual SLAM in dynamic environments has become an urgent and hot issue to be solved in this field in recent years.

In recent years, researchers have proposed several visual SLAM systems to solve the problem of visual SLAM in dynamic environments. The DS-SLAM [3] system proposed by Yu et al. is built on the ORB-SLAM2 [4] framework, and innovatively combines the lightweight semantic segmentation network SegNet [5]

with the optical flow method to construct an efficient motion consistency detection module, which solves the problem that the traditional dynamic elimination algorithm cannot run in real time due to excessive computational overhead. Dai et al. [6] proposed a point correlation-based SLAM algorithm, which clusters point clouds and separates static backgrounds from dynamic objects by analyzing the consistency of 3D spatial point distances between consecutive frames, thus achieving high-precision camera pose estimation in complex environments containing unknown dynamic objects. The paper by Chang [7] introduces a lightweight object detection network, which greatly improves the real-time processing speed of dynamic SLAM while ensuring system accuracy. Fan et al. [8] introduced a lightweight network based on multi-task learning, which can provide object detection bounding boxes and pixel-level semantic masks at real-time frame rate simultaneously, and realize dynamic environment SLAM with both high accuracy and real-time performance combined with multi-view geometric constraints. Wang [9] proposed a highly flexible and modular system framework, which completely decouples the deep learning front-end from the traditional SLAM back-end through Inter-Process Communication (IPC), realizing the plug-and-play of deep learning models in SLAM systems. Aiming at the positioning problems in unstructured scenes, the paper by Liu [10] proposed to segment and eliminate dynamic interferences in real time and further use static scene semantics to enhance the reliability of feature matching, thus achieving robust pose estimation in extreme environments. The paper published by Chen et al. [11] takes ORB-SLAM3 as the benchmark framework, lightens the segmentation model through knowledge distillation technology and combines with the dynamic probability propagation mechanism to improve the system's elimination capability of dynamic feature points.

2. System framework

ORB-SLAM2 is a milestone open-source system in the field of visual SLAM and also the first complete SLAM scheme supporting monocular, stereo and RGB-D cameras. In terms of system architecture, ORB-SLAM2 abandons the traditional filter framework and fully adopts a parallel multi-threaded design based on graph optimization. The core of the system consists of three parallel running threads: Tracking, Local Mapping and Loop Closure Detection. These three threads work collaboratively by maintaining a shared map data structure to realize real-time and high-precision camera localization and sparse map construction.

Based on ORB-SLAM2, the open-source framework of the classic ORB-SLAM series in the feature point method, this paper improves the front-end visual odometry part of this classic visual SLAM framework by combining the instance segmentation-based dynamic point elimination algorithm and the geometry-based dynamic point detection algorithm introduced above. The overall framework of the improved system is shown in Figure 1. In addition to the original threads of ORB-SLAM2, the system adds a dynamic point elimination module, which is mainly used to complete the detection of dynamic regions in images and the filtering of dynamic feature points. It includes an instance segmentation detection thread executed in parallel with other threads, and adopts a mask repair algorithm based on morphological post-processing, which well improves the problem of unstable output masks with defects or omissions of the YOLACT instance segmentation network, obtains a more accurate dynamic object mask, and can significantly improve the accuracy of the dynamic point elimination algorithm based on the instance segmentation network. The other part of the module is dynamic point detection based on dense optical flow, which can not only filter out the feature points on dynamic targets for which the instance segmentation network fails to extract masks, but also filter out the feature points on objects with non-dynamic prior information but actually moving with dynamic targets.

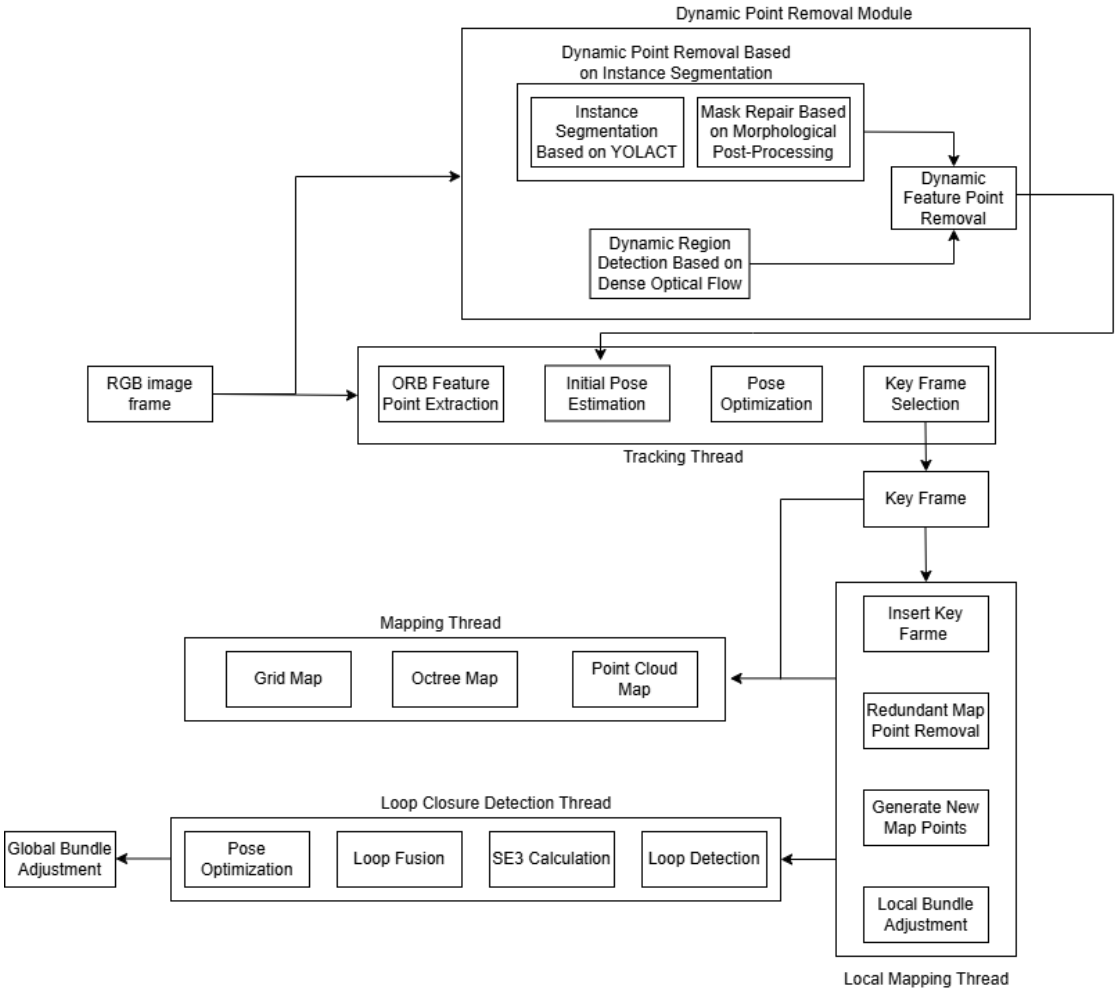

Figure 1. SLAM system framework integrated with the dynamic point elimination module

3. Experiments and analysis

3.1. Dynamic object segmentation based on YOLACT

To strike a balance between segmentation accuracy and inference speed, this paper selects YOLACT [12] as the core instance segmentation network. YOLACT is a groundbreaking fully convolutional one-stage real-time instance segmentation network proposed by Bolya et al. in 2019. YOLACT innovatively decouples the complex instance segmentation task into two highly efficient and fully parallel subtasks, and fuses them through simple matrix operations, thus achieving ultra-fast inference speed while maintaining high-resolution and high-quality mask output.

In the evaluation system of visual SLAM systems, the real-time performance of the system is always a very important key evaluation index. This stringent real-time constraint stems from the typical application scenarios of visual SLAM technology in the real world: whether it is autonomously navigating mobile service robots, lightweight micro unmanned aerial vehicles or resource-constrained autonomous driving edge computing platforms, these devices can usually only be equipped with embedded processors with extremely low power consumption and limited computing capacity. The graphical and computing resources they can provide are far from comparable to the large cluster devices with multiple high-performance GPUs often equipped in

laboratory environments. The pre-trained model with ResNet-50-FPN as the backbone network has the shortest single-frame segmentation time. Considering the real-time performance of the visual SLAM system, this paper adopts the YOLACT-550 pre-trained model trained with ResNet-50-FPN as the backbone network as the benchmark network of the system's instance segmentation module. The images processed by the YOLACT network and the generated masks are shown in Figure 2.

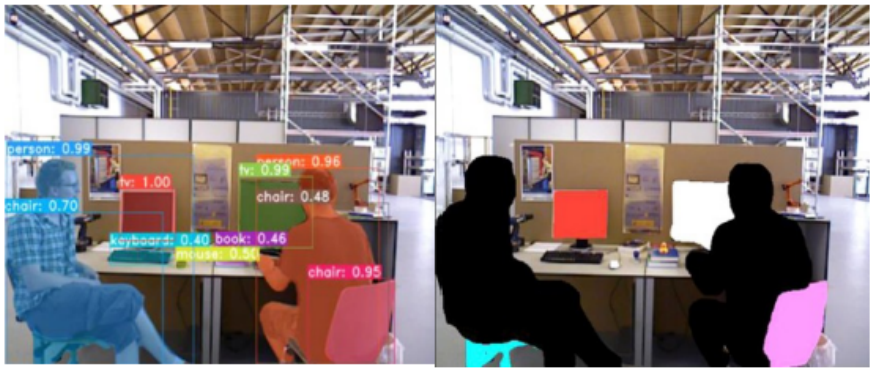

Figure 2. Images processed by YOLACT network and the generated masks

3.2. Dynamic mask optimization based on morphological post-processing

Since YOLACT is trained on the COCO [13] dataset and speed is often inversely proportional to accuracy, the original output mask of YOLACT usually faces four typical problems: (1) zigzag boundary distribution; (2) isolated small noise patches; (3) unactivated holes inside the object mask; (4) structural fragmentation or breakage in the mask of large-scale objects. In visual odometry, a large number of high-quality ORB feature points are often attached to the physical edges of objects or the junctions of internal textures. If these rough and broken original masks are directly used for the judgment of feature points, it is easy to cause false or missing elimination of feature points, thus leading to camera pose drift (see Figure 3 for the unstable mask output of YOLACT).

Figure 3. Unstable mask output of YOLACT

This section designs a set of mask post-processing modules based on image processing and morphology in a targeted manner. The module performs structural repair and boundary refinement on the mask of dynamic targets through low computational cost steps such as thresholding, connected component filtering, opening and closing operations, and Gaussian smoothing in sequence. The complete process is as follows:

(1) Mask binarization and threshold segmentation: The prediction head of the YOLACT network actually outputs a probability confidence map that each pixel belongs to a certain dynamic instance. To clearly define the dynamic region, an empirical threshold T (T = 0.5 is used in this paper) is first set, and the probability map

is converted into a binary mask by using hard threshold segmentation technology. Pixels higher than the threshold are clearly marked as dynamic foreground (assigned a value of 1 or 255), and pixels lower than the threshold are classified as static background (assigned a value of 0).

(2) Connected component analysis and small target elimination: Disturbed by ambient light reflection or similar textures of background clutter, isolated small noise patches are often scattered around the binarized mask. The physical size of these noises is much smaller than that of real dynamic objects (such as pedestrians and vehicles). This step calculates the pixel area of each independent connected region, and the system directly sets the tiny connected regions with an area smaller than the minimum area threshold to 0 by setting the minimum area threshold, thus effectively eliminating the discrete outlier noise patches caused by network misjudgment.

(3) Edge denoising based on opening operation: Aiming at the tiny burrs remaining outside the true contour of the target, the opening operation in mathematical morphology is introduced. The mask is scanned with a structural element of appropriate size (a 3×3 rectangular kernel is used in this paper). The opening operation can smooth the object contour, break extremely narrow false connection bridges, and further eliminate the isolated convex noise on the mask edge without significantly changing the macroscopic area of the target.

(4) Hole filling and structural repair based on closing operation: Aiming at the internal holes caused by insufficient activation of the network's internal texture of objects and the structural fragmentation of the mask of large-scale objects (such as side-parked cars), the morphological closing operation is executed. The closing operation can effectively bridge tiny cracks inside the mask, reintegrate the adjacent but broken mask blocks, fill the unactivated holes inside them, and make the originally broken mask of dynamic objects reclose into a dense and continuous integral instance mask.

(5) Edge softening and smoothing based on Gaussian filtering: After the above morphological repair, the overall structure of the mask tends to be complete and dense, but the binary edge often presents stepped sharp zigzags due to pixel-level erosion and dilation operations. Considering that the feature points at the object edge during SLAM feature extraction are prone to oscillatory misjudgment between static and dynamic states due to pixel jitter. Therefore, the final step introduces Gaussian smoothing filtering to process the hard boundary of the mask. Through convolution with a 2D Gaussian kernel, the originally stiff 0-1 jump boundary transitions more naturally. The comparison of the mask before and after repair is shown in Figure 4.

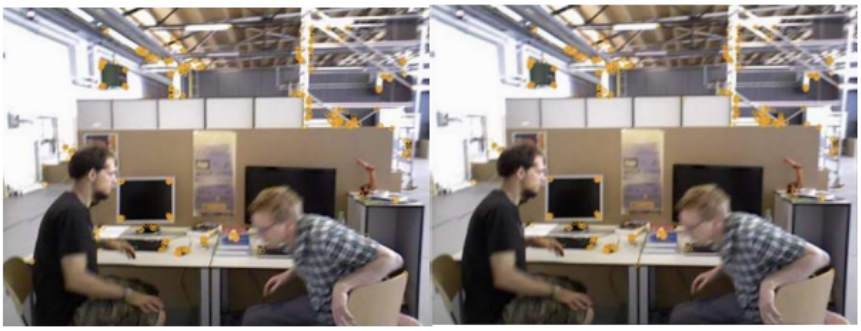

Figure 4. Comparison of the mask before and after repair

3.3. Dynamic region detection based on dense optical flow

This paper adopts the Farneback dense optical flow algorithm proposed by Gunnar Farneback in 2003 for dynamic region detection. The core idea of the algorithm is to approximately fit the gray distribution in the tiny neighborhood of each pixel in the image into a local 2D quadratic polynomial surface. For any pixel point

$x = (x, y)^T$ in the image frame I_1 at time t_1, the gray value in its neighborhood can be approximately expressed by the following quadratic function:

$$I_1(x) = x^T A_1 x + b_1^T x + c_1 \tag{1}$$

where A_1 is a symmetric matrix containing the local second-order derivative information of the image (describing the curvature of the surface); b_1 is a vector containing the local first-order derivative information (gradient); c_1 is a constant term representing the central gray value of the neighborhood. At the subsequent time t_2, assuming that the image I_1 is transformed into the image I_2 through an unknown global or local translation vector $d = (\Delta x, \Delta y)^T$, then according to the brightness constancy assumption, the local gray distribution polynomial of the image I_2 at the same point x can be expressed as the result of the original polynomial translated by *d*:

$$I_2(x) = I_1(x - d) = (x - d)^T A_1 (x - d) + b_1^T (x - d) + c_1 \tag{2}$$

Expand and rearrange it into the standard quadratic form: $I_2(x) = A_2 x + b_2^T x + c_2$. By comparing the coefficients of the two polynomials I_1 and I_2, we can obtain an extremely elegant algebraic relationship. In particular, for the linear term coefficient b, the following equation holds:

$$b_2 = b_1 - 2A_1 d \tag{3}$$

Ideally, as long as the matrix A_1 is non-singular (i.e., invertible), we can directly calculate the pixel displacement vector (optical flow) d by solving this linear equation set:

$$d = -\tfrac{1}{2} A_1^{-1} (b_2 - b_1) \tag{4}$$

This paper directly adopts the Farneback dense optical flow calculation scheme in OpenCV, an open-source computer vision library. At the code implementation level, the system extracts the global dense motion field frame by frame by calling the calcOpticalFlowFarneback function officially provided by OpenCV. The function integrates the complete processes such as image pyramid construction, polynomial expansion and Gaussian neighborhood smoothing inside. The obtained dense optical flow field is shown in Figure 5.

Figure 5. Visualization results of dense optical flow

Experiments on the TUM [14] dataset in this paper show that the dynamic region detection effect is the best when the horizontal and vertical offset threshold of pixel points is set to 5 pixels. The effect of filtering dynamic points on the dynamic region obtained by dense optical flow is shown in Figure 6.

Figure 6. Results of dynamic point filtering based on dense optical flow

3.4. Experimental results

The proposed algorithm is compared with SLAM dynamic scene processing algorithms such as DS-SLAM [15] and DynaSLAM [16]. Table 1 shows the comparison of Absolute Trajectory Error (ATE) between the proposed system and the ORB-SLAM2 system on the fr3 sequence. Table 2 shows the validity verification results of the system module using the fr3_walking_xyz sequence. Table 3 shows the comparison of ATE between the proposed system and the DS-SLAM system on the walking scene sequence. Table 4 shows the comparison of ATE between the proposed system and the DynaSLAM system on the walking scene sequence. The trajectory error comparison diagram of ORB-SLAM2 is shown in Figure 7.

Table 1. Comparison of Absolute Trajectory Error between the proposed system and ORB-SLAM2 on the fr3 sequence (m)

Sequence	ORB-SLAM2			Proposed System			Improvement		
	mean	rmse	std	mean	rmse	std	mean	rmse	std
walking_xyz	0.663	0.786	0.428	0.046	0.015	0.007	93.1%	98.1%	98.4%
walking_half	0.377	0.42	0.161	0.030	0.028	0.015	92.0%	93.3%	90.7%
walking_static	0.351	0.397	0.182	0.019	0.006	0.003	94.6%	98.5%	98.4%
walking_rpy	0.600	0.688	0.333	0.036	0.032	0.020	94.0%	95.3%	94.0%
sitting_xyz	0.012	0.014	0.006	0.009	0.009	0.005	25.0%	35.7%	16.7%
sitting_half	0.034	0.038	0.016	0.016	0.018	0.009	52.9%	52.6%	43.8%
sitting_static	0.009	0.011	0.005	0.007	0.006	0.003	22.2%	45.5%	40.0%
sitting_rpy	0.028	0.039	0.028	0.017	0.02	0.012	39.3%	48.7%	57.1%

Table 2. Validity verification results of the system module using the fr3_walking_xyz sequence

SLAM System	rmse	mean	std	Average Tracking Time
ORB-SLAM2	0.6629	0.7863	0.4279	34.5ms
ORB-SLAM2 + Instance Segmentation	0.0186	0.0161	0.0098	91.2ms
ORB-SLAM2 + Geometric Dynamic Point Detection	0.1512	0.1128	0.1039	60.2ms
The complete proposed system	0.0462	0.0146	0.0071	153.2ms

Table 3. Comparison of Absolute Trajectory Error between the proposed system and DS-SLAM on the walking scene sequence (m)

Sequence	DS-SLAM2		Proposed System		Improvement	
	rmse	std	rmse	std	rmse	std
walking_xyz	0.025	0.016	0.015	0.007	40.0%	56.3%
walking_half	0.034	0.022	0.028	0.015	17.6%	31.8%
walking_static	0.008	0.004	0.006	0.003	25.0%	25.0%
walking_rpy	0.444	0.235	0.032	0.02	92.8%	91.5%

Table 4. Comparison of Absolute Trajectory Error between the proposed system and DynaSLAM on the walking scene sequence (m)

Sequence	Dyna-SLAM2		Proposed System		Improvement	
	rmse	std	rmse	std	rmse	std
walking_xyz	0.017	0.009	0.015	0.007	11.8%	22.2%
walking_half	0.032	0.016	0.028	0.015	12.5%	6.3%
walking_static	0.007	0.003	0.006	0.003	14.3%	0.0%
walking_rpy	0.037	0.022	0.032	0.02	13.5%	9.1%

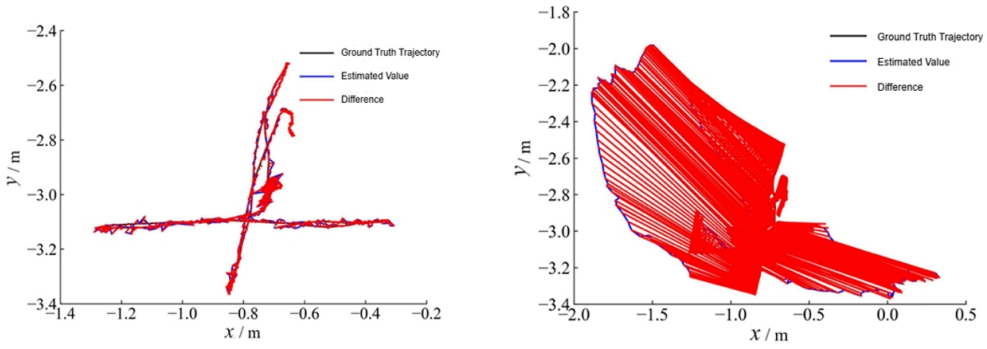

Figure 7. Trajectory error comparison diagram of ORB-SLAM2

4. Conclusion

Aiming at the problem of degraded or even missing segmentation accuracy caused by the use of lightweight instance segmentation networks in the dynamic point elimination algorithm based on instance segmentation networks, a mask repair algorithm based on morphological post-processing is proposed, which solves the above problems well. Then the dynamic point elimination algorithm based on dense optical flow for dynamic region detection is introduced in detail. Next, the specific operation steps of the dynamic point elimination module are designed based on the above algorithms and integrated into the ORB-SLAM2 system framework. Finally, experimental analysis on the TUM dataset verifies that the improved system proposed in this paper can effectively reduce the impact of dynamic objects in the environment on the localization accuracy of the visual SLAM system and improve the system's robustness in dynamic scenes. The comprehensive performance of the proposed system is better than that of ORB-SLAM2, DS-SLAM and DynaSLAM.

References

[1] Cadena, C., Carlone, L., Carrillo, H., & Latif, Y. (2016). Past, present, and future of simultaneous localization and mapping: toward the robust-perception age. *IEEE Transactions on Robotics, 32*(6), 1309–1332.

[2] Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics, 31*(5), 1147–1163.

[3] Yu, C., Liu, Z., Liu, X. J., & Xie, F. (2018). DS-SLAM: A semantic visual SLAM towards dynamic environments. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 1168–1174). IEEE.

[4] Mur-Artal, R., & Tardós, J. D. (2017). ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics, 33*(5), 1255–1262.

[5] Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(12), 2481–2495.

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

[7] Chang, Z. Y., Wu, H. L., Sun, Y. L., & Chen, J. (2022). RGB-D visual SLAM based on Yolov4-tiny in indoor dynamic environment. *Micromachines, 13*(2), 230.

[8] Fan, Y. C., Zhang, Q. C., Tang, Y. L., & Huang, H. (2022). Blitz-SLAM: a semantic SLAM in dynamic environments. *Pattern Recognition, 121*, 108225.

[9] Wang, X., Wang, N., & Zhang, G. (2024). *XRDSLAM: A flexible and modular framework for deep learning based SLAM*. arXiv preprint. https://doi.org/10.48550/arXiv.2410.23690

[10] Yang, L., & Cai, H. (2024). Enhanced visual SLAM for construction robots by efficient integration of dynamic object segmentation and scene semantics. *Advanced Engineering Informatics, 59*, 102313. https://doi.org/10.1016/j.aei.2023.102313

[11] Chen, L., Ling, Z., Gao, Y., & Wang, H. (2023). A real-time semantic visual SLAM for dynamic environment based on deep learning and dynamic probabilistic propagation. *Complex & Intelligent Systems, 9*(4), 5653–5677.

[12] Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019). YOLACT: real-time instance segmentation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 9157–9166). IEEE.

[13] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer vision – ECCV 2014* (Vol. 8693, pp. 740–755). Springer. https://doi.org/10.1007/978-3-319-10602-1_48

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

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

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