IL-SLAM: Intelligent Line-assisted SLAM Based on Feature Awareness for Dynamic Environments

Haolan Zhang[1], Thanh Nguyen Canh[1], Chenghao Li[1], Ruidong Yang[1], Yonghoon Ji[1], and Nak Young Chong[1,2]
[1]Graduate School of Advanced Science and Technology, Japan Advanced Institute of Science and Technology, Nomi, Japan
[2]Department of Robotics, Hanyang University, Ansan, Korea
{haolan.z, thanhnc, chenghao.li, yang_ruidong, ji-y, nakyoung}@jaist.ac.jp

***Abstract*—Visual SLAM is crucial for autonomous systems but struggles in dynamic environments. While recent dynamic SLAM systems use geometric constraints and deep learning to remove dynamic features, this creates a new challenge: insufficient remaining point features. Existing solutions continuously introduce additional line and plane features, causing unnecessary computational overhead and potential performance degradation from low-quality features. We propose IL-SLAM with a feature-aware mechanism that quantitatively evaluates point feature adequacy (abundance and distribution) to selectively activate line features only when necessary. This mathematically-grounded decision process minimizes computational complexity while reducing additional noise introduction. Our hierarchical optimization strategy uses line features for tracking and local mapping to improve initial pose estimation, but excludes them from global optimization to prevent long-term drift from low-quality additional features. Extensive experiments on TUM datasets demonstrate substantial improvements in both ATE and RPE metrics compared to ORB-SLAM3 baseline and superior performance over other dynamic SLAM and multi-feature methods.**

***Index Terms*—Visual SLAM, Feature-aware mechanism, Line-assissted, Dynamic environments.**

I. Introduction

Visual SLAM is crucial for autonomous mobile systems. Traditional feature-based methods [1]–[3] excel in texture-rich static environments due to high computational efficiency, but their static assumptions cause pose estimation failures in dynamic scenarios. To address this, approaches evolved from geometric outlier rejection [4]–[6] to hybrid methods combining deep learning and geometry for dynamic feature removal [7]–[9]. While improving robustness, this creates a critical challenge: after aggressive dynamic removal, remaining point features are not always sufficient for reliable SLAM processing.

Recent multi-feature dynamic SLAM methods address feature insufficiency through additional features. In PLD-SLAM [10], the authors combine RGB-D images with semantic segmentation to identify predefined dynamic regions, then utilize K-means [11] clustering to remove potential dynamic point and line features, finally employing joint point-line optimization for robust pose estimation. DRG-SLAM [12] introduces line and plane features alongside points, making comprehensive use of environmental geometric information. It employs SegNet [13] and epipolar constraints for predefined dynamic object removal, while using multi-view constraints for undefined dynamics. The system adaptively selects between Manhattan World Pose Estimation and standard methods based on scene structure. PLDS-SLAM [14] improves line segment matching through geometric constraints while employing Bayesian algorithms to continuously track and remove dynamic noise from both points and lines. YPL-SLAM [15] extracts line features within static regions obtained after epipolar-based dynamic removal for pose estimation.

While these methods effectively supplement features in dynamic environments, they adopt continuous introduction of additional features regardless of actual necessity, even when point features remain sufficient in static or low-dynamic scenarios. This causes two critical problems: (i) unnecessary computational overhead from processing additional features in stable conditions, and (ii) accumulation of low-quality features and noise over time, adversely affecting both pose estimation accuracy and long-term global optimization.

To address these problems, we propose IL-SLAM with a feature-aware mechanism that quantitatively evaluates point feature adequacy to selectively introduce line features only when necessary. Our approach minimizes computational overhead while maintaining tracking robustness through quantitative feature quality assessment and hierarchical optimization. The key contributions are as follows.

1) A feature-aware additional feature introduction mechanism that ensures minimal computational overhead by introducing line features only when point features are insufficient, avoiding unnecessary processing in scenarios where point features are adequate.
2) A minimal line feature processing strategy that participates in tracking and local mapping for pose correction but excludes line features from global optimization to prevent low-quality feature interference with long-term accuracy.
3) Extensive comparative experiments on TUM [16] datasets demonstrating significant improvements over state-of-the-art dynamic SLAM methods and trajectory error reduction compared to our baseline ORB-SLAM3.

This work was supported in part by the Asian Office of Aerospace Research and Development under Grant/Cooperative Agreement Award No. FA2386-25-1-4034, and in part by JST SPRING, Japan Grant Number JPMJSP2102.

II. METHODOLOGY

The framework of our proposed method is shown in Fig. 1. RGB-D images serve as input for ORB feature extraction and YOLOV8 [17] object detection. Dynamic removal is performed on point features using detected predefined dynamic regions and geometric constraints. The remaining point features are evaluated by our feature-aware mechanism to determine whether to activate point-line mode. When activated, line features are extracted using LSD [18], undergo dynamic removal with the same potential dynamic regions, and participate in tracking with points for initial pose estimation. Finally, pose optimization uses points alone to prevent interference from additional features.

A. Feature-Aware Mechanism

Inspired by texture richness evaluation in multi-feature SLAM [19], [20], we design a quantitative mechanism to assess point feature adequacy after dynamic removal. Our goal is to determine when remaining point features are sufficient for robust tracking by evaluating two critical factors: (i) feature abundance: whether enough features remain, and (ii) feature distribution: whether features are evenly spread across the image. Insufficient abundance causes tracking failure from too few correspondences, while poor distribution leads to pose estimation degeneracy in certain image regions.

To quantify these factors, we divide each frame into $G = 9$ (3×3) grids and evaluate each grid i independently. For abundance, we normalize the feature count c_i by a baseline c_{base}, yielding c_i/c_{base}. For distribution, we penalize spatial variance σ_i^2 using $1/(1+\sqrt{\sigma_i^2})$, where higher variance (clustered features) produces lower scores. Combining both aspects and averaging over all grids yields our feature quality metric:

$$Q_{feature} = \frac{1}{G}\sum_{i=1}^{G}\left(\frac{c_i}{c_{base}} + \frac{1}{1+\sqrt{\sigma_i^2}}\right) \tag{1}$$

The decision to activate line features is:

$$\text{Scene} = \begin{cases} \textbf{Point} & \text{if } Q_{feature} \geq th \\ \textbf{Point-Line} & \text{otherwise} \end{cases} \tag{2}$$

where th is the classification threshold. Both c_{base} and th are derived from stable datasets (fr3/sitting/static in TUM and BONN synchronous1,2) and set to minimum values to ensure conservative classification, only activating line features when point features are clearly insufficient. This mechanism minimizes computational overhead while reducing low-quality feature interference.

B. Line Extraction and Matching

When the feature-aware mechanism determines that additional features are needed for supplementation (point-line mode), we employ LSD [18] to extract line features. For the line features, following the representation method from [12], they are represented in 3D space as $L = (S_L, Q_L^1, M_L, Q_L^2, E_L)$, where these correspond to the start point, quarter point 1, midpoint, quarter point 2, and end point, respectively. Their projection on the image plane is denoted as $l = (s_l, q_l^1, m_l, q_l^2, e_l)$. For subsequent line segment matching, based on the above representation, we perform line feature matching with descriptor. Considering the auxiliary role of line features, the matching process adopts a simple matching approach. Given descriptors (ℓ_i, A_i, R_i) and (ℓ_j, A_j, R_j) representing length, angle, and edge strength response for line segments l_i and l_j respectively, the matching distance is computed by combining length difference, angle difference, edge strength difference. A line match is accepted when the distance is below a threshold and passes the ratio test.

In the local mapping stage, we employ our search-projection function to associate maplines with current frame observations using the same descriptor distance metric defined in line feature matching, where 3D maplines are projected into the image plane and matched within a spatial search window.

C. Dynamic removal

For dynamic removal, We adopt an aggressive dynamic removal strategy to maximize elimination of dynamic interference. This conservative approach validates our feature-aware mechanism's effectiveness even when point features are minimized. We first utilize YOLOV8 detection results to remove features within predefined dynamic object regions. Point features P_p falling within these regions are directly removed. For line features, if three or more sampling points $(S_L, Q_L^1, M_L, Q_L^2, E_L)$ fall within dynamic regions, the entire line is removed, following [12].

After completing the first step removal based on detection results, we apply epipolar constraints to remove undefined dynamic features on the remaining features. For point features P_p, we compute their distance d to the epipolar line and remove those exceeding threshold $d > 1$ (following [7]). For line features, we compute the epipolar distance for each sampling point individually. Following the same rule as detection-based removal, if three or more sampling points exceed the threshold, the entire line is removed.

D. Pose Estimation

When line features are activated, both point and line features participate in the tracking and local mapping processes through reprojection error minimization.

For point features, the reprojection error is computed as:

$$e_P^i = \|u_i - \pi(K \cdot T_{cw} \cdot P_i^w)\|^2 \tag{3}$$

For line features, the reprojection error is based on the distance between projected 3D line endpoints and observed 2D line endpoints:

$$e_L^i = \|s_l^i - \pi(K \cdot T_{cw} \cdot S_L^{w,i})\|^2 + \|e_l^i - \pi(K \cdot T_{cw} \cdot E_L^{w,i})\|^2 \tag{4}$$

where u_i is the observed 2D point coordinates, P_i^w is the corresponding 3D world point, s_l^i, e_l^i and $S_L^{w,i}$, $E_L^{w,i}$ are observed/world coordinates of line endpoints of line i, T_{cw} is the camera pose, K is the camera intrinsic matrix, and $\pi(\cdot)$ represents the projection function.

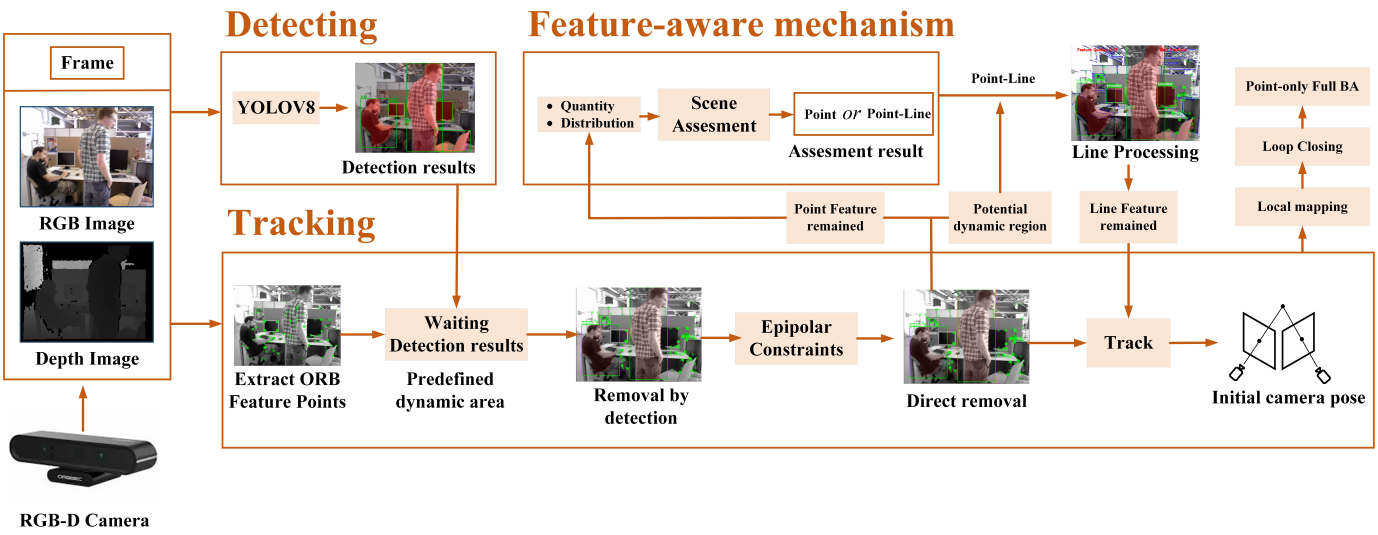

Fig. 1. **Overview of our proposed IL-SLAM framework:** The framework is composed of three main units: feature-aware mechanism, Detecting and Tracking.

The pose estimation minimizes the combined reprojection error:

$$T_{cw}^{*} = \arg\min_{T_{cw}} \left(\sum_{i} \rho(e_P^i) + \sum_{i} \rho(e_L^i) \right) \tag{5}$$

where $\rho(\cdot)$ is a robust kernel function. Following our hierarchical optimization strategy, both point and line features participate in tracking and local bundle adjustment for accurate initial poses. However, global bundle adjustment uses only point features to prevent low-quality additional features from interfering with long-term map consistency.

III. EXPERIMENTAL RESULTS

To comprehensively evaluate and validate our IL-SLAM method, we conducted extensive experiments on TUM [16] RGB-D datasets, encompassing diverse camera motion patterns including hemispherical, XYZ translation, RPY rotation, and static configurations across various indoor scenarios. We employ absolute trajectory error (ATE) and relative pose error (RPE) as primary metrics. The root mean square error (RMSE) and standard deviation (S.D.) are utilized to characterize both trajectory accuracy and system stability. The RPE evaluation, relative translation error (T.RPE) to provide comprehensive pose estimation analysis. All experiments were conducted on Ubuntu 20.04 with AMD Ryzen R7 CPU, NVIDIA RTX3080 GPU, and 32GB RAM.

Fig. 2 demonstrates the performance of our method on both a low dynamic sequence (fr3/w/static) and a complex dynamic sequence (fr3/w/xyz). From left to right, the upper two images and the lower two images represent the current frame visualization and mapping results for fr3/w/static and fr3/w/xyz, respectively. In the low-dynamic scenario (fr3/w/static), the first image shows the detection results with green feature points extracted from the current frame. The green text in the upper portion evaluates the current feature quality, providing a good feature assessment. The corresponding second image displays the sparse mapping results, retaining non-dynamic red map points and green map lines. The blue and red boxes represent the estimated camera poses.

In contrast, fr3/w/xyz contains significant dynamic objects that are correctly detected and removed from human regions. After dynamic removal, the poor feature quality assessment (red text indicating "bad feature") triggers our system to activate point-line mode, introducing line features (blue lines) for supplementation. The sparse mapping shows notably more map lines compared to fr3/w/static. This visualization demonstrates our feature-aware mechanism's adaptability: efficiently using point features in stable scenarios while automatically supplementing with line features when dynamic removal causes feature insufficiency.

For validation, we comprehensively selected five representative sequences: fr3/w/half, fr3/w/rpy, fr3/w/xyz, fr3/s/xyz, and fr3/w/static. The first three feature different camera motions (translational, rotational, combined) with continuous human movement. fr3/s/xyz has seated persons with camera translation, while fr3/w/static involves minimal dynamics. These sequences cover diverse combinations of camera and environmental dynamics.

First, we compare our method with our baseline ORB-SLAM3 [3] and several state-of-the-art dynamic SLAM methods from recent years: Blitz-SLAM [9], RTD-SLAM [21] (referred to as RTD), and COEB-SLAM [22] (referred to as COEB). Blitz-SLAM employs semantic segmentation combined with epipolar geometry, RTD-SLAM integrates object detection with multi-view geometry, and COEB-SLAM combines semantic segmentation with optical flow information. The comparison results with these methods are presented in Table I. Based on the overall results across the tested

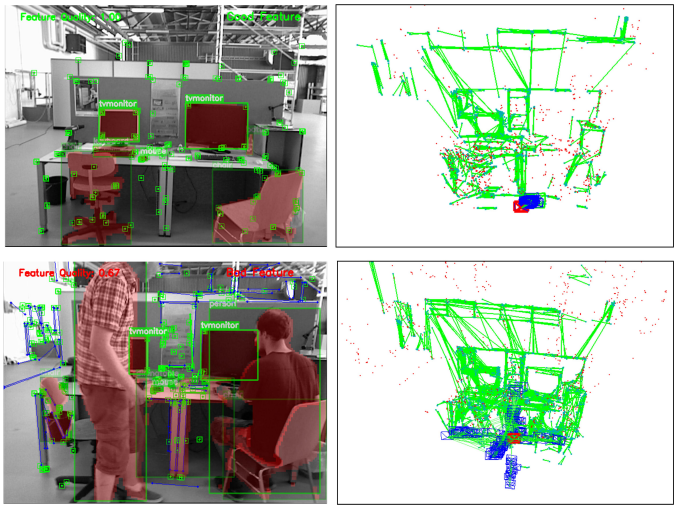

Fig. 2. Performance visualization of our feature-aware mechanism on fr3/w/static (upper two images) and fr3/w/xyz (lower two images) sequences. The system adaptively switches between point mode in stable scenarios and point-line mode in complex dynamic environments based on feature quality assessment.

datasets, our method demonstrates outstanding performance, achieving the best results (highlighted in bold) in both ATE and T.RPE metrics across all five test sequences, consistently outperforming our baseline ORB-SLAM3 and other state-of-the-art dynamic SLAM methods.

In complex dynamic sequences (fr3/w/half, fr3/w/rpy, fr3/w/xyz), our method achieves best performance. For example, in fr3/w/rpy, our ATE RMSE of 0.032 significantly outperforms ORB-SLAM3's 0.160, validating our aggressive dynamic removal strategy. In low-dynamic scenarios (fr3/w/static, fr3/s/xyz), our method maintains optimal performance with ATE RMSE of 0.007 in fr3/w/static, demonstrating our feature-aware mechanism's ability to compensate for potential over-removal through selective line feature supplementation.

The T.RPE results further validate the robustness of our approach, with our method consistently achieving the best translational accuracy across all sequences. For example, in fr3/w/half, our T.RPE of 0.016 outperforms all competing methods, while in fr3/w/xyz, we achieve 0.011 compared to ORB-SLAM3's 0.027. These results confirm that our feature-aware mechanism effectively maintains accurate pose estimation under varying environmental dynamics while minimizing computational complexity through selective feature introduction.

Additionally, we compared our feature-aware SLAM with several multi-feature systems to highlight its advantages over continuous feature integration. O3L extends ORB-SLAM3 with line features, enabling direct comparison of selective versus continuous line use. Planar [23], DRG-SLAM [12], and YPL-SLAM [15] combine point, line, and (for some) plane features, all robust to dynamic interference, making them suitable baselines. Table II summarizes the results.

In ATE evaluation, the fr3/s/xyz sequence mainly involves camera motion with minimal dynamics, where our selective strategy yields a slightly lower RMSE. Across four dynamic sequences, our method achieves the best performance, with only a 0.001 deviation from the top result in fr3/w/half, confirming the effectiveness of selective feature use and dynamic removal in complex scenes.

For RPE, our pose continuity is slightly inferior due to aggressive dynamic filtering, but remains close to the best performer O3L, demonstrating that selective feature introduction preserves trajectory consistency while minimizing interference, validating the effectiveness of our feature-aware design.

Fig. 3 and Fig. 4 show trajectory and RPE comparisons between our method and ORB-SLAM3 across five sequences. ORB-SLAM3 fails in the first four dynamic sequences due to motion interference, with RPE errors of 0.35–1.4. In contrast, our IL-SLAM achieves stable estimation with RPE errors within 0.06–0.2, confirming effective dynamic handling. In the final sequence with minimal dynamics, both methods exhibit nearly identical trajectories and RPE around 0.06, validating that our selective feature processing preserves baseline accuracy under stable conditions.

Finally, Table III shows IL-SLAM incurs moderate overhead (0.0586 s/frame vs ORB-SLAM3's 0.0363 s/frame). Critically, IL-SLAM(-F) without feature-aware mechanism requires 0.0718 s/frame, confirming our selective introduction strategy reduces computational cost by 18%, validating our design goal of minimizing unnecessary processing.

IV. CONCLUSIONS

This paper presents IL-SLAM, a feature-aware line-assisted SLAM system that adaptively introduces line features only when point features are insufficient after dynamic removal. This selective mechanism addresses two key issues in existing multi-feature SLAM: redundant computation from continuous feature use and accuracy degradation caused by low-quality additional features.

Experiments on the TUM RGB-D datasets show that IL-SLAM consistently outperforms ORB-SLAM3 and other dynamic SLAM systems, achieving the best ATE results in four out of five sequences. The proposed hierarchical optimization, where line features contribute to local mapping but are excluded from global bundle adjustment, effectively balances robustness and long-term accuracy.

Future work will focus on replacing the current threshold-based decision with a more comprehensive feature evaluation inspired by QualiSLAM [24], enabling joint assessment of point and auxiliary features from a unified quality perspective to further improve the balance between efficiency and accuracy.

REFERENCES

[1] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015.

TABLE I
COMPARISON BETWEEN OUR IL-SLAM AND THE LATEST DYNAMIC SLAM SYSTEMS

Sequences	ABSOLUTE TRAJECTORY ERROR (ATE/m)									
	ORB-SLAM3		Blitz-SLAM		RTD		COEB		OURS	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
fr3/w/half	0.231	**0.009**	**0.025**	0.012	0.028	0.024	0.028	0.014	**0.025**	0.014
fr3/w/rpy	0.160	0.073	0.035	0.022	0.167	0.030	0.033	0.020	**0.032**	**0.019**
fr3/w/static	0.024	0.012	0.010	0.005	0.121	**0.002**	**0.007**	0.003	**0.007**	0.003
fr3/w/xyz	0.275	0.145	**0.015**	**0.007**	0.020	0.009	0.016	0.008	**0.015**	**0.007**
fr3/s/xyz	**0.012**	0.006	0.014	0.006	-	-	-	-	**0.012**	**0.005**
	METRIC TRANSLATIONAL DRIFT (T.RPE/rad)									
fr3/w/half	0.024	0.016	0.025	0.012	0.035	0.024	0.032	0.017	**0.016**	**0.011**
fr3/w/rpy	0.030	0.021	0.047	0.028	**0.019**	**0.013**	0.046	0.027	0.026	0.020
fr3/w/static	0.019	0.016	0.012	0.006	0.019	0.013	0.009	**0.003**	**0.006**	**0.003**
fr3/w/xyz	0.027	0.020	0.019	0.009	0.012	0.007	0.021	0.011	**0.011**	**0.006**
fr3/s/xyz	**0.009**	0.006	0.014	0.007	-	-	-	-	0.010	**0.005**

Note: The best results of RMSE and S.D. are highlighted in bold, and the second best are underlined.

TABLE II
COMPARISON BETWEEN OUR IL-SLAM AND THE EXISTING SLAM SYSTEMS USING PLANES OR LINES AS FEATURES

Sequences	ABSOLUTE TRAJECTORY ERROR (ATE/m)									
	O3L		Planar		DRG-SLAM		YPL-SLAM		OURS	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
fr3/w/half	0.209	0.095	0.325	-	**0.025**	-	0.027	**0.013**	**0.025**	0.014
fr3/w/rpy	0.158	0.077	0.553	-	0.385	-	0.044	0.025	**0.032**	**0.019**
fr3/w/static	0.020	0.011	0.293	-	**0.007**	-	0.009	0.004	**0.007**	**0.003**
fr3/w/xyz	0.276	0.119	0.276	-	0.018	-	0.026	0.014	**0.015**	**0.007**
fr3/s/xyz	0.009	**0.005**	0.024	-	**0.008**	-	-	-	0.012	**0.005**
	METRIC TRANSLATIONAL DRIFT (T.RPE/rad)									
fr3/w/half	**0.010**	**0.007**	0.051	-	**0.010**	-	0.023	0.012	0.016	0.011
fr3/w/rpy	**0.010**	**0.006**	0.051	-	0.042	-	0.044	0.030	0.026	0.020
fr3/w/static	**0.004**	**0.003**	0.023	-	0.004	-	0.009	0.004	0.006	**0.003**
fr3/w/xyz	**0.009**	**0.006**	0.036	-	0.009	-	0.019	0.009	0.011	**0.006**
fr3/s/xyz	**0.006**	**0.003**	0.009	-	0.007	-	-	-	0.010	0.005

Note: The best results of RMSE and S.D. are highlighted in bold, and the second best are underlined.

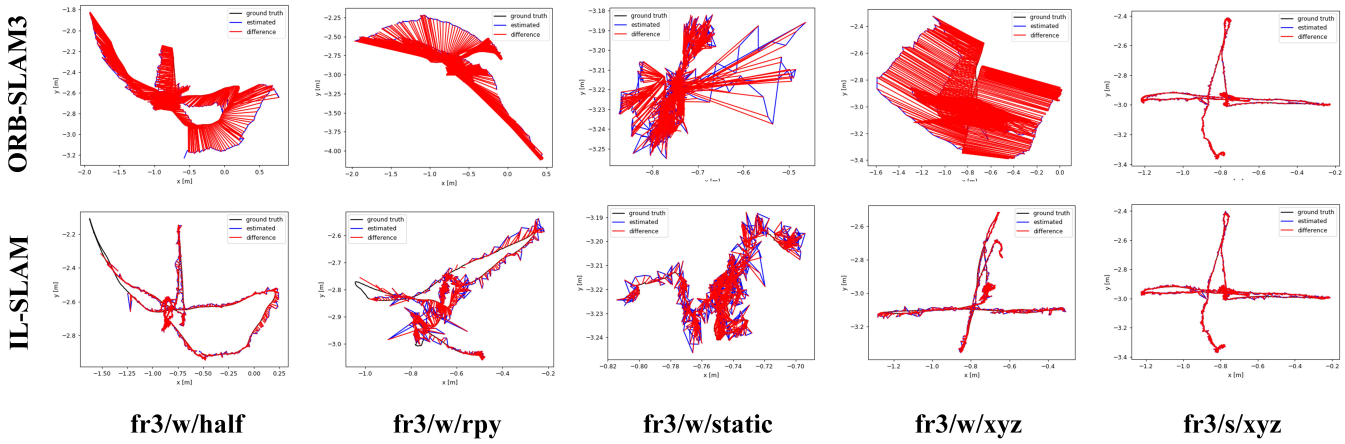

Fig. 3. ATE results of IL-SLAM and ORB-SLAM3 running in five sequences, where the black line represents the true trajectory, the blue line represents the trajectory estimated by the algorithm, and the red line represents the difference between the estimated and true values

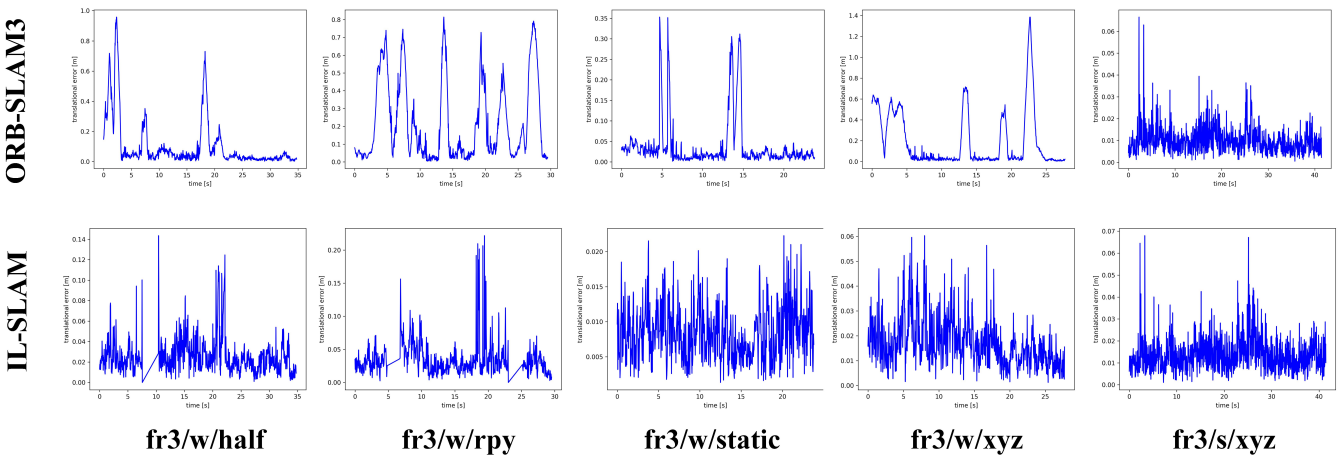

Fig. 4. RPE results of IL-SLAM and ORB-SLAM3 running in five sequences, where the blue lines represent the RPE results for each time point.

TABLE III
OVERALL RUNTIME ANALYSIS

Sequences	Overall Runtime (frame/s)		
	ORB-SLAM3	IL-SLAM	IL-SLAM(-F)
fr3/w/half	0.037	0.057	0.073
fr3/w/rpy	0.037	0.056	0.071
fr3/w/static	0.036	0.063	0.071
fr3/w/xyz	0.036	0.057	0.072
fr3/s/xyz	0.036	0.060	0.072
Average	0.0363	0.0586	0.0718

[2] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source slam system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[3] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[4] Y. Sun, M. Liu, and M. Q.-H. Meng, "Improving RGB-D SLAM in dynamic environments: A motion removal approach," *Robot. Auton. Syst.*, vol. 89, pp. 110–122, 2017.

[5] S. Li and D. Lee, "RGB-D SLAM in dynamic environments using static point weighting," *IEEE Robot. Automat. Lett.*, vol. 2, no. 4, pp. 2263–2270, Oct. 2017

[6] W. Dai, Y. Zhang, P. Li, Z. Fang, and S. Scherer, "RGB-D SLAM in dynamic environments using point correlations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 373–389, Jan. 2022.

[7] C. Yu et al., "DS-SLAM: A semantic visual SLAM towards dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.(IROS)*, Oct. 2018, pp. 1168–1174

[8] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 4076–4083, Oct. 2018.

[9] Y. Fan, Q. Zhang, Y. Tang, S. Liu, and H. Han, "Blitz-SLAM: A semantic SLAM in dynamic environments," *Pattern Recognit.*, vol. 121, Jan. 2022, Art. no. 108225.

[10] Zhang, C.; Huang, T.; Zhang, R.; Yi, X. PLD-SLAM: A New RGB-D SLAM Method with Point and Line Features for Indoor Dynamic Scene. *ISPRS Int. J. Geo-Inf.* 2021, 10, 163. https://doi.org/10.3390/ijgi10030163

[11] G. Hamerly and C. Elkan, "Learning the k in k-means," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 16, 2003, pp. 281–288.

[12] Y. Wang, K. Xu, Y. Tian, and X. Ding, "DRG-SLAM: A semantic RGB-D SLAM using geometric features for indoor dynamic scene," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 1352–1359

[13] 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, Dec. 2017.

[14] Yuan, C.; Xu, Y.; Zhou, Q. PLDS-SLAM: Point and Line Features SLAM in Dynamic Environment. *Remote Sens*. 2023, 15, 1893. https://doi.org/10.3390/rs15071893

[15] Du, X.; Zhang, C.; Gao, K.; Liu, J.; Yu, X.; Wang, S. YPL-SLAM: A Simultaneous Localization and Mapping Algorithm for Point–line Fusion in Dynamic Environments. *Sensors* 2024, 24, 4517. https://doi.org/10.3390/s24144517

[16] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2012, pp. 573–580.

[17] R. Varghese and M. Sambath, "YOLOv8: A novel object detection algorithm with enhanced performance and robustness," in *Proc. Int. Conf. Adv. Data Eng. Intell. Comput. Syst. (ADICS)*, Apr. 2024, pp. 1–6, doi: 10.1109/ADICS58448.2024.10533619.

[18] R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A Fast Line Segment Detector with a False Detection Control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 722–732, Apr. 2010.

[19] Rui Gao, Shuo Wu, Liyang Zhang, Lei Pan, Guoju Zhang, Hongjie Wang, and Xiaojing Wang. Plfg-slam: A visual slam for indoor weak-texture envi- ronments with point-line feature glue matching. *Measurement*, 256:118435, 2025.

[20] Yan, J., Zheng, Y., Yang, J., Mihaylova, L., Yuan, W. and Gu, F. (2024) PLPF-VSLAM: An indoor visual SLAM with adaptive fusion of point-line-plane features. *Journal of Field Robotics*, 41, 50–67. https://doi.org/10.1002/rob.22242

[21] R. Gao, Z. Li, J. Li, B. Li, J. Zhang, and J. Liu, "Real-time SLAM based on dynamic feature point elimination in dynamic environment," *IEEE Access*, vol. 11, pp. 113952–113964, 2023.

[22] F. Min, Z. Wu, D. Li, G. Wang, and N. Liu, "COEB-SLAM: A robust VSLAM in dynamic environments combined object detection, epipolar geometry constraint, and blur filtering," *IEEE Sensors J.*, vol. 23, no. 21, pp. 26279–26291, Nov. 2023.

[23] Y. Li, R. Yunus, N. Brasch, N. Navab, and F. Tombari, "RGB-D SLAM with structural regularities," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 11581–11587.

[24] Z. Ma, D. He, X. Gao, Y. Yang, W. Yu and T. -K. Truong, "QualiSLAM: Quality-Driven Feature Adaptation for Visual SLAM in Challenging Environments," *IEEE Trans. Instrum. Meas.*, vol. 74, pp. 1–16, 2025.