Scientific Reports

https://doi.org/10.1038/s41598-026-46629-0

Article in Press

Enhanced visual-inertial SLAM Using SuperPoint and semantic geometric dynamic feature detection

Received: 24 November 2025

Accepted: 26 March 2026

Published online: 31 March 2026

Cite this article as: Cui J., Huang Y. & Wang L. Enhanced visual-inertial SLAM Using SuperPoint and semantic geometric dynamic feature detection. *Sci Rep* (2026). https://doi.org/10.1038/s41598-026-46629-0

Jianyuan Cui, Yingping Huang & Lele Wang

Enhanced Visual-Inertial SLAM Using SuperPoint and Semantic Geometric Dynamic Feature Detection

Jianyuan Cui[1], **Yingping Huang**[1,*], **and Lele Wang**[2]

[1]School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai, 200093, China
[2]School of Artificial Intelligence, Suzhou Chien-Shiung Institute of Technology, Taicang, 215411, China
[*]980485655@qq.com

ABSTRACT

Visual–inertial simultaneous localization and mapping (VI-SLAM) is fundamental for unmanned driving and VR. However, traditional feature-based SLAM systems rely on hand-crafted features and lack dedicated methods for handling dynamic objects, which results in degraded performance under challenging conditions, such as violent motion, varying illumination, and dynamic environments. To handle the issues, we propose SuperDynaSLAM, an enhanced VI-SLAM that integrates SuperPoint which is a deep learning–based feature extractor with a two-stage dynamic feature point detection method. By replacing the traditional ORB extractor with SuperPoint, SuperDynaSLAM can extract more robust feature points under challenging conditions. Furthermore, by fusing semantic information and geometric constraints, SuperDynaSLAM can accurately detect moving objects and remove associated dynamic feature points. Experiments conducted on multiple datasets demonstrate that SuperDynaSLAM achieves more competitive performance compared with ORB-SLAM3 and other SLAM systems.

Introduction

Simultaneous Localization and Mapping (SLAM) plays a pivotal role in various domains, including unmanned driving and augmented reality[1–3]. It enables a device to estimate its own trajectory while concurrently constructing a map of the surrounding environment in previously unknown spaces.

In feature-based SLAM systems, the robustness of the front-end feature extractor is crucial for overall stability and accuracy. Some systems such as ORB-SLAM[4] use the ORB feature point[5] for its efficiency, but this design sacrifices robustness[6]. For example, violent motion acts as a low-pass filter that smooths intensity gradients and suppresses high-frequency image components, directly weakening the FAST corner detector used by the ORB extractor[7]. As a result, the number and repeatability of extracted feature points are reduced, leading to mismatching and unstable pose estimation.

Furthermore, most existing SLAM systems, such as ORB-SLAM3[8], lack dedicated methods for handling dynamic objects, even though such objects are common in real world and can introduce dynamic feature points. Due to their independent motion, the dynamic feature points introduce incorrect geometric constraints and must be removed to prevent tracking error. While Random Sample Consensus (RANSAC)[9] can reject a small number of outliers, its effectiveness diminishes when the dynamic objects dominate the scene. Moreover, the dynamic feature points also increases bundle adjustment complexity and interferes with loop closure. Therefore, detecting and removing dynamic feature points is essential to achieve robust and efficient SLAM performance[10].

To address the issues mentioned above, we propose SuperDynaSLAM, a VI-SLAM that integrates SuperPoint[11] which is a deep learning–based feature extractor with a dynamic feature point detection method that fuses semantic information and geometric constraints. Replacing the ORB detector with SuperPoint enables SuperDynaSLAM to produce more robust feature points under challenging conditions. Semantic masks generated by Mask R-CNN[12] classify feature points into background, vehicle, and pedestrian classes, and matching is performed only within each class to reduce cross-class mismatching. Since not all movable objects contained in the semantic masks are moving currently, SuperDynaSLAM further introduces the geometric constraint from the IMU measurements to identify moving objects. The IMU measurements provide short-term motion estimates, from which a fundamental matrix is computed to achieve the expected epipolar line of static feature points. Feature points from moving objects tend to deviate significantly from their expected epipolar lines. By measuring the distance between feature point and its expected epipolar line, SuperDynaSLAM can accurately identify moving objects and remove the associated dynamic feature points, thereby improving overall localization accuracy. The main contributions of this article are summarized as follows:

1. We replace the ORB extractor with SuperPoint, providing SuperDynaSLAM with uniform distributional and robust

feature points. This substantially enhances tracking stability under challenging conditions. And the feature points extracted by SuperPoint are divided into three classes based on the semantic information from Mask R-CNN to avoid mismatching.

2. We design and implement a dynamic feature point detection method. By fusing semantic information provided by Mask R-CNN with geometric constraints from the IMU, SuperDynaSLAM can accurately detect moving objects in the scene, thereby enabling the removal of associated dynamic feature points.

3. We evaluated SuperDynaSLAM on three public VI-SLAM datasets, EuRoC, VIODE and OpenLORIS-Scene. The experimental results show that SuperDynaSLAM achieves performance improvements compared with ORB-SLAM3 and other SLAM methods.

Related work

To overcome the challenges that dynamic environments pose to SLAM, researchers have proposed a variety of solutions. These methods can be broadly categorized into three groups: geometry-based methods, semantic-based methods, and hybrid methods that fuse geometric and semantic information.

Geometry-based methods

Early SLAM methods primarily relied on geometric constraints to identify and remove dynamic feature points. The core assumption of these methods is that points on the static parts of a scene must adhere to strict epipolar or motion model constraints across different frames, whereas dynamic feature points will violate those constraints[13]. For instance, the ORB-SLAM3 treats dynamic feature points as outliers and employs RANSAC during tracking to filter out observations that are inconsistent with the static model. While effective in scenes with low dynamics, this passive outlier rejection would fail if moving objects dominate the field of view.

To more proactively handle dynamic feature points, Jaimez et al.[14] proposed a strategy based on geometric clustering. This method clusters the 3D point cloud into multiple rigid groups and segments the scene into a dynamic foreground and a static background based on their motion residuals, using only the static background for camera pose estimation. A different clustering method was taken by Song et al.[15] in DGM-VINS, which applies DBSCAN clustering within a 2D feature space defined by vector distance and epipolar constraints to identify dynamic feature outliers. Other works have explored the temporal consistency of scene structure. Dai et al.[16] introduced the concept of "point correlation" by constructing a sparse graph with Delaunay Triangulation and culling edges that exhibit inconsistent relative positions over time. Similarly, BaMVO[17] builds a non-parametric background model from a sequence of depth images, classifying pixels as dynamic if their current depth measurement significantly deviates from the established static model. These methods are advantageous as they require no prior object knowledge but can be challenged by slow or camera-consistent motions. Beside that, InertialNet[18] introduces an end-to-end learning-based approach that predicts IMU rotation directly from image sequences via optical flow, providing a robust inertial prior for visual–inertial motion estimation under challenging conditions.

Semantic-based methods

With the advancement of deep learning, leveraging semantic information to identify dynamic objects has become a mainstream method[19]. These methods use object detection or instance segmentation networks to directly identify object categories that typically exhibit mobility, such as pedestrians and vehicles, at the image level and exclude their feature points from the SLAM backend.

Zhong et al.[20] designed an efficient strategy in Detect-SLAM that runs an object detector only on keyframes and mitigates detector latency by propagating a moving probability for regular frames. Hu et al.[21], in their CFP-SLAM, adopted a coarse-to-fine probabilistic framework that not only detects objects using YOLOv5 but also calculates a static probability for each object to distinguish between high and low dynamic properties, thus avoiding the excessive removal of static features. DynaSLAM[22] uses the semantic segmentation results of Mask R-CNN to detect dynamic content in a complementary fashion. The DOTMask framework by Vincent et al.[23] takes this a step further by using a YOLACT instance segmentation network[24] to generate pixel-wise masks and designing a dual-masking strategy: one for creating a clean, static map and another for the real-time frontend odometry. These methods can effectively handle known categories of dynamic objects, but their performance is limited by the accuracy of the detection or segmentation network.

Hybrid methods fusing geometry and semantics

To combine the strengths of the two aforementioned categories, recent research has focused on hybrid methods that fuse geometric constraints with semantic information. These hybrid methods typically employ semantic information for initial

guidance or hypothesis generation, which is subsequently verified and refined through geometric constraints. In this way, they achieve robust performance in handling dynamic objects.

Yan et al.[25] designed a semantic frame selection strategy in DGS-SLAM, running a lightweight segmentation network only when necessary, and fusing its results with a K-Means clustering-based motion residual model via an adaptive threshold. SG-SLAM[26] obtains the semantic information of dynamic objects through the object detection network. Under the premise of maintaining accuracy, the speed of the algorithm is greatly improved through the combination of appropriate corrections and the ORB-SLAM2[27]. More recently, RSO-SLAM, proposed by Qin et al.[28], integrates three sources of information: semantic segmentation, dense optical flow, and sparse feature matching. This method utilizes a k-means and connectivity algorithm to cluster the optical flow field for pure geometric motion cues, which are then merged with semantic masks. Finally, a precise motion probability is calculated via an optical flow attenuation propagation strategy. These hybrid methods generally outperform single-modality methods in both accuracy and robustness. They therefore represent the current frontier of dynamic SLAM research.

Methods

System overview

SuperDynaSLAM is an enhanced VI-SLAM built upon ORB-SLAM3 that integrates SuperPoint which is a deep learning–based feature extractor with a two-stage dynamic feature point detection method. As shown in Fig. 1, SuperDynaSLAM replaces the traditional ORB extractor with SuperPoint, which provides more robust and more spatially uniform feature points. To further improve performance in dynamic environments, SuperDynaSLAM introduces a two-stage dynamic feature point detection method that fuses semantic segmentation and geometric constraints. Except for these front-end enhancements, the remaining components of SuperDynaSLAM, including local mapping, loop closing, and back-end optimization, are directly inherited from ORB-SLAM3 and remain unchanged.

Given an input image, Mask R-CNN is first used to generate an initial semantic mask that identifies all movable objects, such as vehicles and pedestrians. The same image is simultaneously processed by SuperPoint to extract feature points, which are classified into background, vehicle, and pedestrian classes based on their locations within the initial mask. Feature matching is then performed within each class to prevent cross-class mismatches. The detailed procedure is described in Section SuperPoint-based feature extraction and matching method.

Although the initial mask contains all movable objects, not all of them are actually moving. To distinguish the moving objects, SuperDynaSLAM incorporates geometric constraints from the IMU measurements. By computing the distance between feature points and their corresponding epipolar lines derived from the IMU measurements, SuperDynaSLAM can further distinguish the moving objects from the movable ones which have already been identified in the initial mask. This enables SuperDynaSLAM to remove dynamic feature points with higher precision. The detailed procedure is described in Section Two-stage dynamic feature point detection method fusing semantic information and geometric constraints.

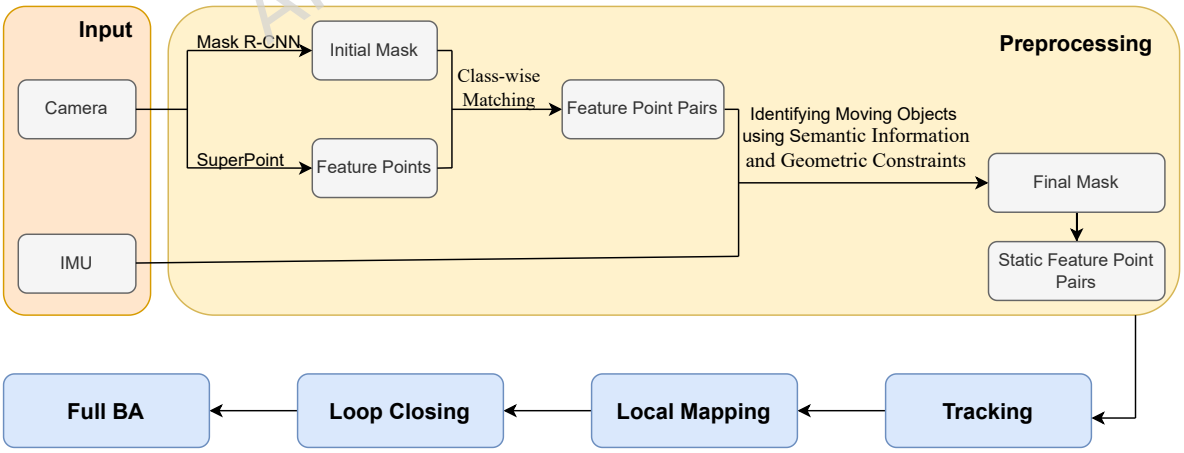


Figure 1. Overall framework of SuperDynaSLAM.

SuperPoint-based feature extraction and matching method

To enhance the robustness of the SLAM system, SuperDynaSLAM replaces the traditional ORB extractor with SuperPoint, a deep learning–based feature extraction network. Unlike the ORB extractor, which relies on hand-crafted features, SuperPoint

learns to extract feature points through self-supervised training. As illustrated in Fig. 2, SuperPoint is a fully-convolutional neural network comprising a shared encoder and two parallel decoder heads. The encoder uses VGG-style convolutional layers and spatial downsampling to produce compact feature maps. One decoder head detects keypoint locations, while the other computes the corresponding descriptors.

The training process of SuperPoint consists of three stages. First, a base detector named MagicPoint is trained on synthetic images containing simple geometric shapes with no ambiguity in the keypoint locations. Then, Homographic Adaptation which warps the real-world image multiple times is used in conjunction with MagicPoint to generate the pseudo-ground truth keypoints. Finally, SuperPoint is trained on warped real-world images using the pseudo-ground truth keypoints, enabling it to jointly extract keypoints and descriptors from an image.

The core of SuperPoint is Homographic Adaptation which can apply several random homographies including translation, scale, in-plane rotation, and symmetric perspective distortion to the original image. With the warped images, the feature extractor can see the scene from many different viewpoints and scales, which gives a large boost in robustness especially under challenging conditions. Moreover, because Homographic Adaptation can be randomly sampled and applied to any input image, it effectively increases the number of training samples available for SuperPoint network.

When SuperPoint operates as a feature extractor in SuperDynaSLAM, the input image is first processed by the shared encoder to obtain a set of dense feature maps, which are then fed into two decoder heads: one for detecting keypoints and the other for generating the corresponding descriptors. Compared with hand-crafted features, the resulting feature points exhibit improved repeatability, spatial distribution, and matching reliability.

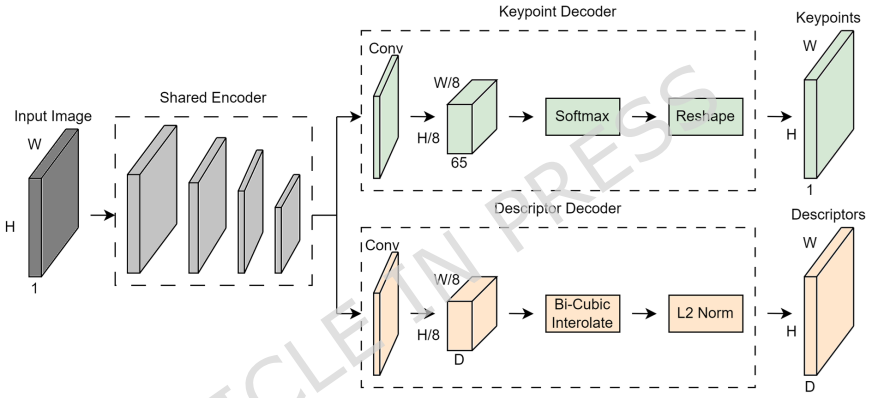


Figure 2. Neural network architecture of SuperPoint.

After the feature points are extracted, SuperDynaSLAM uses the semantic masks generated by Mask R-CNN to classify them into three classes based on their locations: vehicle feature points, pedestrian feature points, and background feature points.

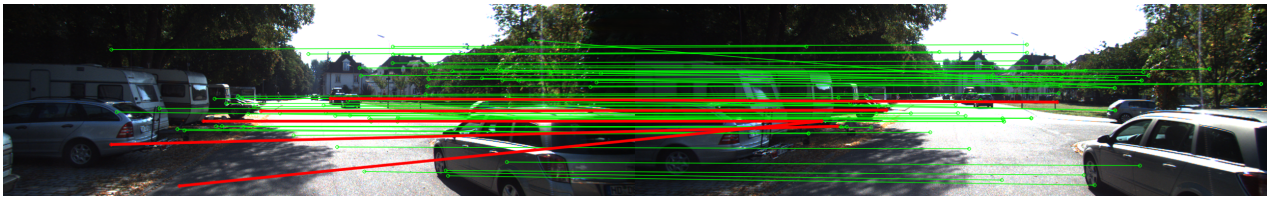
During feature matching, a correctly matched background feature point should correspond to another background feature point. If a background feature point matches a vehicle or pedestrian feature point, the correspondence is considered erroneous. As shown in Fig. 3a, where the original images are from the KITTI dataset[29], several background feature points located on buildings are incorrectly matched with vehicle feature points. These mismatches are highlighted in red.

To address this issue, we propose a class-wise matching strategy. Using the semantic mask obtained from Mask R-CNN, all feature points are divided into three aforementioned classes. During feature matching, semantic constraints are introduced so that similarity is computed only between feature points belonging to the same class, thereby preventing cross-class matching. As illustrated in Fig. 4, background feature points are compared exclusively with other background feature points, while vehicle and pedestrian feature points are matched within their respective classes. As seen in Fig. 3b, this class-wise matching strategy effectively eliminates cross-class mismatches.

Two-stage dynamic feature point detection method fusing semantic information and geometric constraints

The proposed dynamic feature point detection method consists of two stages. In the first stage, Mask R-CNN detects all movable objects in the original image and generates an initial mask. Although the initial mask contains all movable objects, some of them may remain static at a given moment, such as parked vehicles or stationary pedestrians. To address this limitation, our method introduces geometric constraints derived from the IMU measurements to identify the moving objects from those movable objects in the second stage. The overall workflow of the proposed two-stage dynamic feature point detection method is illustrated in Fig. 5.

In the first stage, Mask R-CNN generates an initial mask that contains all movable objects, such as vehicles and pedestrians,



(a) Feature matching without semantic constraints, where cross-class mismatches (highlighted in red) frequently occur.

(b) Feature matching with class-wise semantic constraints, where cross-class mismatches are suppressed.

Figure 3. Illustration of cross-class feature mismatches and their suppression using class-wise matching.

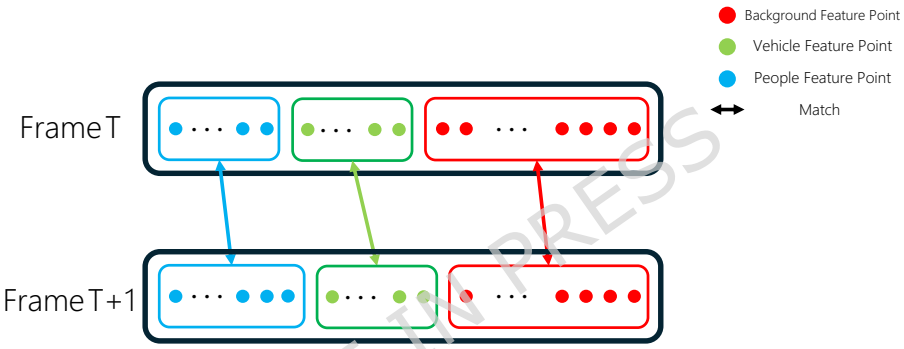


Figure 4. Class-wise matching strategy.

within the original image, as shown in Fig. 7b. The Mask R-CNN model uses the pre-trained weights provided by PyTorch, thereby avoiding the need for retraining and significantly reducing both time and computational cost.

In the second stage, for each movable object, our method only needs to select a small subset of the feature points located on the object to examine whether they satisfy the geometric constraints, rather than using all of them, to determine whether the object is moving. The IMU measurements are used by our method to construct geometric constraints that enable reliable discrimination between static and moving objects. Let P_{t-1} denote the camera pose at the previous frame. The IMU measurements between the previous and current frames can be obtained as follows:

$$ {}^{b}\widetilde{\omega}(t) = {}^{b}\omega(t) + b_g(t) + \eta_g(t) \tag{1} $$

$$ {}^{b}\widetilde{a}(t) = {}^{W}_{b}R^{T}\left({}^{b}a(t) - {}^{W}g\right) + b_a(t) + \eta_a(t) \tag{2} $$

where ${}^{b}\widetilde{\omega}(t)$ and ${}^{b}\widetilde{a}(t)$ represent the measurements of angular velocity and acceleration; ${}^{b}\omega(t)$ and ${}^{b}a(t)$ represent the true value of angular velocity and acceleration; $b_g(t)$ and $b_a(t)$ represent the gyroscope and accelerometer biases; $\eta_g(t)$ and $\eta_a(t)$ represent the gyroscope and accelerometer random noise; ${}^{W}g$ represents the gravitational acceleration, and ${}^{W}_{b}R^{T}$ represents the rotation matrix respectively. From these IMU measurements, the current pose P_t and the relative transformation matrix $T\left\{R^{3\times3}\middle|t^{3\times1}\right\}$ between two frames can be estimated. The essential matrix E is then computed from the equation 3:

$$ E = [t]_X R \tag{3} $$

Figure 5. Overall workflow of the two-stage dynamic feature point detection method.

where $[t]_X$ is the skew-symmetric matrix of the translation matrix t. With the essential matrix E, the fundamental matrix F is given by the equation 4:

$$F = K^{-T} E K^{-1} \tag{4}$$

where K is the camera intrinsic parameters. As shown in Fig. 6b, for each movable object, a pair of feature points p_1, p_2 is randomly selected. The homogeneous coordinates of the feature points p_1 and p_2 are defined as P_1 and P_2, as follows:

$$P_1 = [x_1, y_1, 1]^T, \; P_2 = [x_2, y_2, 1]^T \tag{5}$$

With the fundamental matrix F, the epipolar line L_2 corresponding to p_1 in the second frame is computed as:

$$L_2 = [X, Y, Z]^T = FP_1 = F\,[x_1, y_1, 1]^T \tag{6}$$

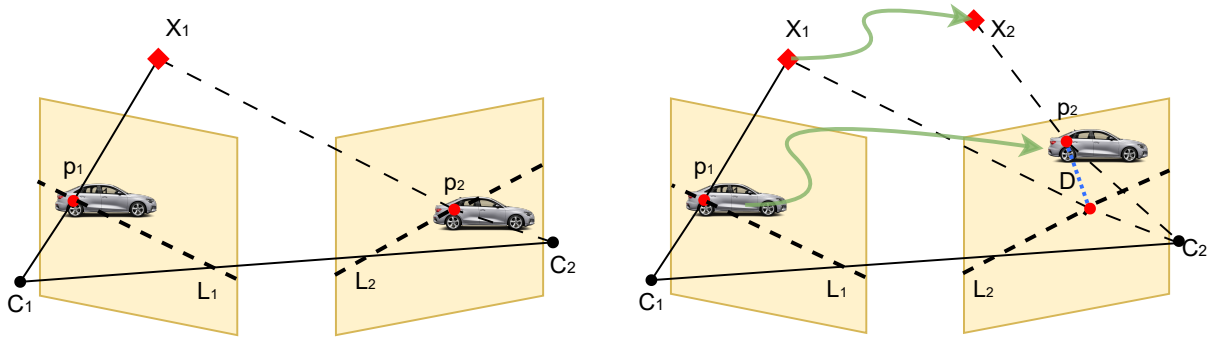
and the offset distance D between p_2 and L_2 be calculated from the equation 7:

$$D = \frac{|P_2 F P_1|}{\sqrt{(X^2 + Y^2)}} \tag{7}$$

If the movable object is stationary, the offset distance should be zero under ideal conditions. As shown in Fig. 6a, when the vehicle is static, the feature point p_2 located on the vehicle will lie exactly on its corresponding epipolar line L_2, meaning that the offset distance between the feature point and the epipolar line is zero.

However, due to noise and other uncertainties, the offset distance is usually non-zero. Therefore, our proposed method defines an adaptive threshold T based on the average offset distance of background feature points which are assumed to be static. If the offset distance of any selected feature point pair within a movable object exceeds the threshold T, the corresponding object is identified as moving; otherwise, it is identified as static.

For example, in Fig. 6b, because the offset distance between the feature point p_2 and its corresponding epipolar line L_2 which is shown as the blue dashed line D exceeds the threshold, the vehicle is identified as a moving object and all the feature points located in it are regarded as dynamic feature points.



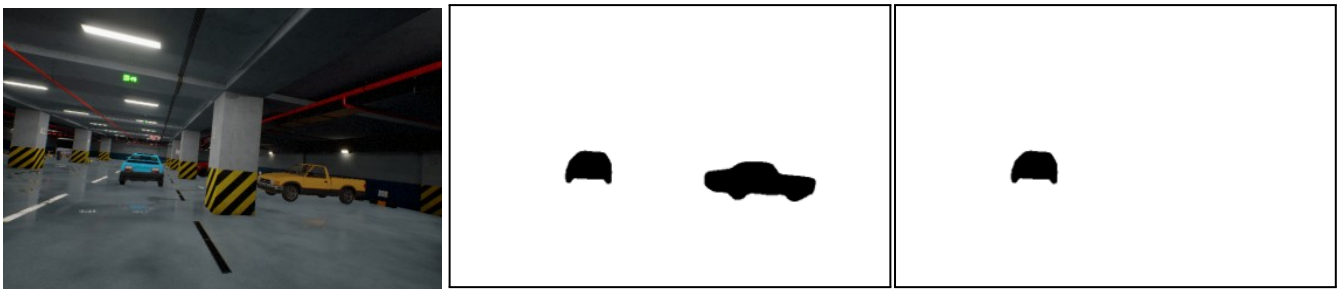
(a) Epipolar consistency observed for a static object.

(b) Epipolar inconsistency caused by a moving object.

Figure 6. Illustration of epipolar consistency and inconsistency for static and moving objects.

After identification, movable objects in the initial mask are separated into moving and static groups. The final mask retains only moving objects, while static ones are removed. Examples of the original image, the initial mask, and the final mask are

shown in Fig. 7. In the original image, there are two movable vehicles, both detected by Mask R-CNN and contained in the initial mask. However, only the vehicle on the left is moving, whereas the one on the right remains stationary. Directly applying the initial mask would incorrectly remove feature points on the right static vehicle. Our method successfully identifies that only the left vehicle is moving and generates a final mask that preserves only this moving object. With the final mask, all dynamic feature points located on moving objects can be accurately detected and removed.



(a) Original image **(b)** Initial mask **(c)** Final mask

Figure 7. Illustration of motion detection.

By fusing semantic information and geometric constraints, the proposed two-stage method achieves efficient detection of moving objects, particularly in scenes where moving objects occupy large portions of the image. This is because our method evaluates only a limited subset of feature points rather than processing all feature points.

Experiments

To verify the effectiveness of SuperDynaSLAM, extensive experiments were conducted on the widely used datasets EuRoC[30], VIODE[31] and OpenLORIS-Scene[32]. The experiments consist of four components. First, the accuracy experiments are designed to assess the overall performance of the proposed method. Second, the ablation studies aim to analyze the contribution of each individual module. Third, a running time analysis is conducted to evaluate the computational cost of different components. Finally, failure cases are analyzed to highlight the limitations of the proposed method and to provide insights into potential directions for future improvement.

In both the accuracy experiments and the ablation studies, we used two commonly adopted metrics, the absolute trajectory error (ATE) and the relative pose error (RPE) proposed in the paper[33]. ATE measures the global discrepancy between the estimated trajectory and the ground-truth trajectory, thereby reflecting the overall positioning accuracy of the SLAM system. The Root Mean Square Error (RMSE) of ATE, denoted as AT-RMSE, is commonly used for quantitative comparison across methods. RPE, on the other hand, quantifies the local pose estimation error between consecutive frames, and characterizes the short-term tracking accuracy of the SLAM system. It includes the translational error (RPEt) and the rotational error (RPEr).

Following the recommendations from the paper[27], each sequence was run five times, and the median results were reported. For ATE, we further report the 95% confidence interval (95% CI) to quantify the statistical variability across five runs.

Accuracy experiments

EuRoC dataset evaluation

The EuRoC dataset is one of the most widely used benchmarks for evaluating VI-SLAM systems. It provides stereo image sequences and synchronized IMU measurements recorded in indoor environments that include diverse and challenging conditions such as violent camera motion, varying illumination, and texture variations. Each of the 11 sequences contains high-precision ground-truth trajectories obtained from a laser-tracking system, enabling rigorous quantitative evaluation of localization and mapping accuracy.

Our method is evaluated and compared with three SLAM methods including the ORB-SLAM3, the PIPO-SLAM[34] and the VINS-Fusion[35]. All the methods are run in the stereo-inertial mode.

The quantitative results of AT-RMSE are summarized in Table 1 and Table 2. According to Table 1, our method achieves lower AT-RMSE values compared with the VINS-Fusion, the PIPO-SLAM and the ORB-SLAM3 in 5 sequences (Seq. MH04, MH05, V102, V103, and V203). Moreover, the Table 2 shows that the 95% CI widths of our method are consistently no larger than those of ORB-SLAM3, indicating more stable localization performance.

Even though most EuRoC sequences contain only a few moving objects, our method still yields higher accuracy than ORB-SLAM3 in ten out of eleven sequences, primarily due to the more uniform spatial distribution and better robustness of the feature points extracted by the SuperPoint. For instance, in the Seq. MH05, the AT-RMSE of our method is 6.30% lower

than that of ORB-SLAM3. However, in relatively stable environments, such as MH01 where illumination is constant and camera motion is moderate, the performance advantage becomes less significant, and in rare cases, ORB-SLAM3 may slightly outperform our method.

Both VINS-Fusion and PIPO-SLAM primarily focus on back-end optimization while still relying on the traditional hand-crafted features. Consequently, under challenging conditions of violent motion or varying illumination, their performance degrades compared with our method, which benefits from the robustness of the feature points extracted by SuperPoint. For example, in the Seq. V103 which involves violet motion and significant viewpoint changes, the AT-RMSE of our method is 4.33% lower than PIPO-SLAM and 53.13% lower than VINS-Fusion. This improvement is mainly attributed to the fact that the feature points extracted by SuperPoint are more robust, as Homographic Adaptation enables SuperPoint to learn to extract keypoints of the same scene under multiple viewpoints.

Seq.	VINS-Fusion	PIPO-SLAM	ORB-SLAM3	SuperDynaSLAM
MH01	0.0852	0.0352	**0.0323**	0.0327
MH02	0.0966	**0.0337**	0.0384	0.0373
MH03	0.1448	**0.0460**	0.0474	0.0471
MH04	0.2147	0.0547	0.0509	**0.0489**
MH05	0.1635	0.0569	0.0603	**0.0565**
V101	0.0660	**0.0835**	0.0851	0.0849
V102	0.1023	0.0623	0.0635	**0.0612**
V103	0.1280	0.0626	0.0641	**0.0600**
V201	0.1920	**0.0509**	0.0514	0.0511
V202	0.2785	**0.0490**	0.0500	0.0495
V203	0.2061	0.0676	0.0668	**0.0648**

Table 1. AT-RMSE [m] (median)

Seq.	VINS-Fusion	PIPO-SLAM	ORB-SLAM3	SuperDynaSLAM
MH01	x	x	0.0331±0.0011	**0.0338±0.0010**
MH02	x	x	**0.0390±0.0008**	0.0371±0.0008
MH03	x	x	0.0469±0.0013	**0.0475±0.0012**
MH04	x	x	**0.0511±0.0024**	0.0465±0.0024
MH05	x	x	0.0585±0.0040	**0.0559±0.0010**
V101	x	x	0.0860±0.0105	**0.0856±0.0095**
V102	x	x	0.0636±0.0069	**0.0603±0.0050**
V103	x	x	0.0645±0.0053	**0.0589±0.0040**
V201	x	x	0.0507±0.0047	**0.0512±0.0037**
V202	x	x	0.0494±0.0064	**0.0492±0.0580**
V203	x	x	0.0677±0.0094	**0.0639±0.0071**

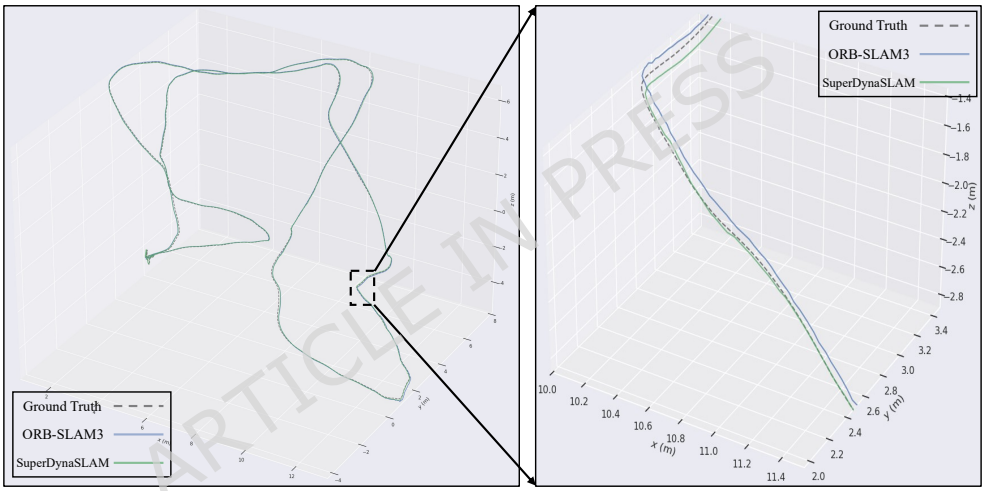
Table 2. AT-RMSE [m] (mean±95% CI)
("x" represents the method does not provide this value in its paper)

The qualitative comparison of trajectories between ORB-SLAM3 and our method is shown in Fig. 8. The trajectories estimated by our system are closer to the ground truth, confirming the superior accuracy and robustness of the proposed method. This observation is consistent with the quantitative AT-RMSE results presented in Table 1 and Table 2.

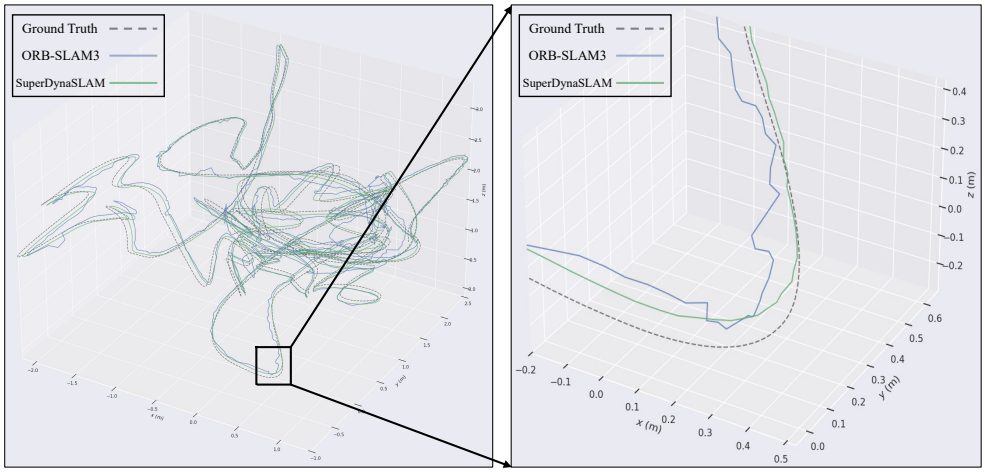
The results of RPE are listed in Table 3. Similar to the AT-RMSE results, our method achieves competitive or superior performance in both RPEt and RPEr components. Because ORB-SLAM3 relies on hand-crafted ORB features, its performance tends to degrade in the presence of violent motion and varying illumination, leading to higher short-term tracking errors. In contrast, our proposed method maintains stable performance under these conditions.

Seq.	VINS-Fusion		PIPO-SLAM		ORB-SLAM3		SuperDynaSLAM	
	RPEt	RPEr	RPEt	RPEr	RPEt	RPEr	RPEt	RPEr
MH01	x	x	x	x	**0.0291**	0.0219	0.0295	**0.0214**
MH02	x	x	x	x	0.0309	**0.0214**	**0.0294**	0.0220
MH03	x	x	x	x	0.0738	0.0322	**0.0728**	**0.0306**
MH04	x	x	x	x	0.0744	0.0250	**0.0716**	**0.0241**
MH05	x	x	x	x	0.0680	0.0237	**0.0638**	**0.0216**
V101	x	x	x	x	**0.0275**	0.0292	0.0277	**0.0289**
V102	x	x	x	x	0.0570	0.0609	**0.0552**	**0.0584**
V103	x	x	x	x	0.0611	0.0797	**0.0565**	**0.0736**
V201	x	x	x	x	0.0512	0.0471	**0.0498**	**0.0467**
V202	x	x	x	x	0.0499	0.0639	**0.0468**	**0.0615**
V203	x	x	x	x	0.2585	0.1673	**0.2422**	**0.1550**

Table 3. The RPEt [m/m] and RPEr [°/m]
("x" represents the method does not provide this value in its paper)



(a) Seq. MH05



(b) Seq. V103

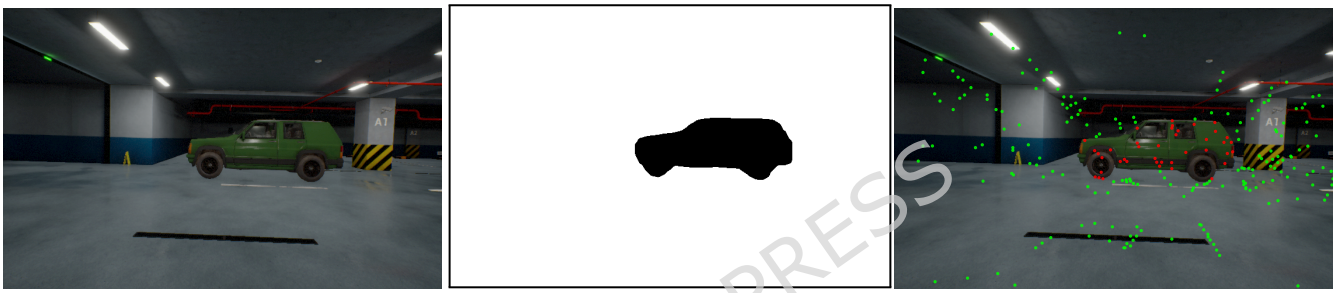
Figure 8. Trajectories of the Seq. MH05(a) and V103(b) estimated by ORB-SLAM3 and ours.

VIODE dataset evaluation

The VIODE dataset is a simulated benchmark specifically designed to evaluate the robustness of VI-SLAM methods in dynamic environments. It contains synchronized stereo RGB images, IMU measurements, and ground-truth trajectories recorded in three representative environments, namely city_day, city_night, and parking_lot. Each environment includes four dynamic levels, namely 0_none, 1_low, 2_mid, and 3_high, corresponding to increasing numbers of moving vehicles. This dataset presents challenging scenarios involving large moving objects and severe occlusions, enabling comprehensive evaluation of SLAM performance in dynamic scenes.

Our method is evaluated and compared with three SLAM methods including the ORB-SLAM3, the DynaSLAM, and the DynaVINS[36]. The ORB-SLAM3, the DynaVINS, and our method are run in stereo-inertial mode. The DynaSLAM is run in stereo mode.

To evaluate the effectiveness of the proposed method in detecting dynamic features, the classification results of static and dynamic feature points are analyzed. As shown in Fig. 9, the proposed two-stage dynamic feature point detection method is able to identify dynamic feature points.



(a) Original image **(b)** Final mask **(c)** Classification result

Figure 9. Detecting dynamic feature points. Dynamic feature points are marked in red and static ones are marked in green.

The quantitative results of AT-RMSE are summarized in Table 4 and Table 5. According to Table 4, our method achieves lower AT-RMSE compared with ORB-SLAM3, DynaSLAM, and DynaVINS in 2 sequences (Seq. parking_lot_1_low and parking_lot_2_mid) and shows a consistent performance gain across nearly all dynamic levels. Similar to the results on the EuRoC dataset, Table 5 shows that our method consistently yields lower 95% CI widths than ORB-SLAM3 across all sequences.

Compared with ORB-SLAM3, our method achieves significant improvements in all sequences. In Seq. city_day_3_high, which contains numerous moving vehicles, our method improves accuracy by 14.42% relative to ORB-SLAM3. These results demonstrate that the combination of the SuperPoint-based feature extraction and matching method and Two-stage dynamic feature point detection method effectively suppresses the adverse impact of dynamic objects.

Across all city_day sequences, our method consistently outperforms DynaSLAM in terms of accuracy. DynaSLAM relies on semantic information to remove feature points located on potentially movable objects, without considering whether these objects are actually moving at the current time. Compared with DynaVINS, our algorithm achieves higher accuracy in two sequences. This improvement arises because DynaVINS removes dynamic feature points based on geometric constraints, making it susceptible to both false removals and missed detections.

The qualitative comparison of trajectories between ORB-SLAM3 and our method is shown in Fig. 10. The trajectory estimated by our system is significantly closer to the ground truth than that of ORB-SLAM3, confirming the improvement in localization accuracy.

The results of RPE are listed in Table 6. Our proposed method achieves lower RPEt and RPEr than ORB-SLAM3 on all sequences except Seq. city_day_0_none and parking_lot_1_low where there are only a few moving objects. In dynamic scenes, ORB-SLAM3 often drifts because moving objects introduce false feature correspondences that degrade geometric consistency, thereby increasing both RPEt and RPEr. In contrast, ours mitigates this issue through the two-stage dynamic feature points detection method fusing semantic information and geometric constraints to remove dynamic feature points accurately. As a result, only static feature points are retained, enabling accurate and robust pose tracking in dynamic environments.

Seq.	Dyna. Level	DynaSLAM	DyanVINS	ORB-SLAM3	SuperDynaSLAM
city_day	0_none	1.735	**0.221**	0.420	0.398
	1_low	1.644	**0.303**	0.649	0.604
	2_mid	1.783	**0.254**	0.410	0.373
	3_high	2.547	**0.277**	0.513	0.439
city_night	0_none	3.758	**0.415**	1.325	1.209
	1_low	3.556	**0.401**	1.303	1.205
	2_mid	4.121	**0.457**	1.208	1.101
	3_high	4.517	**0.524**	1.934	1.662
parking_lot	0_none	**0.207**	0.215	0.260	0.241
	1_low	0.295	0.228	0.227	**0.196**
	2_mid	0.343	0.241	0.259	**0.228**
	3_high	0.395	**0.266**	0.446	0.383

Table 4. AT-RMSE [m] (median)

Seq.	Dyna. Level	DynaSLAM	DyanVINS	ORB-SLAM3	SuperDynaSLAM
city_day	0_none	x	x	0.462±0.0050	**0.415±0.0009**
	1_low	x	x	0.655±0.0149	**0.619±0.0091**
	2_mid	x	x	0.404±0.0139	**0.366±0.0101**
	3_high	x	x	0.523±0.0485	**0.457±0.0422**
city_night	0_none	x	x	1.315±0.0504	**1.200±0.0484**
	1_low	x	x	1.327±0.1186	**1.278±0.0699**
	2_mid	x	x	**1.201±0.0930**	1.225±0.0930
	3_high	x	x	2.008±0.2476	**1.835±0.1878**
parking_lot	0_none	x	x	0.261±0.0042	**0.236±0.0037**
	1_low	x	x	0.231±0.0111	**0.199±0.0041**
	2_mid	x	x	0.255±0.0080	**0.218±0.0043**
	3_high	x	x	0.450±0.0486	**0.392±0.0394**

Table 5. AT-RMSE [m] (mean±95% CI)
("x" represents the method does not provide this value in its paper)

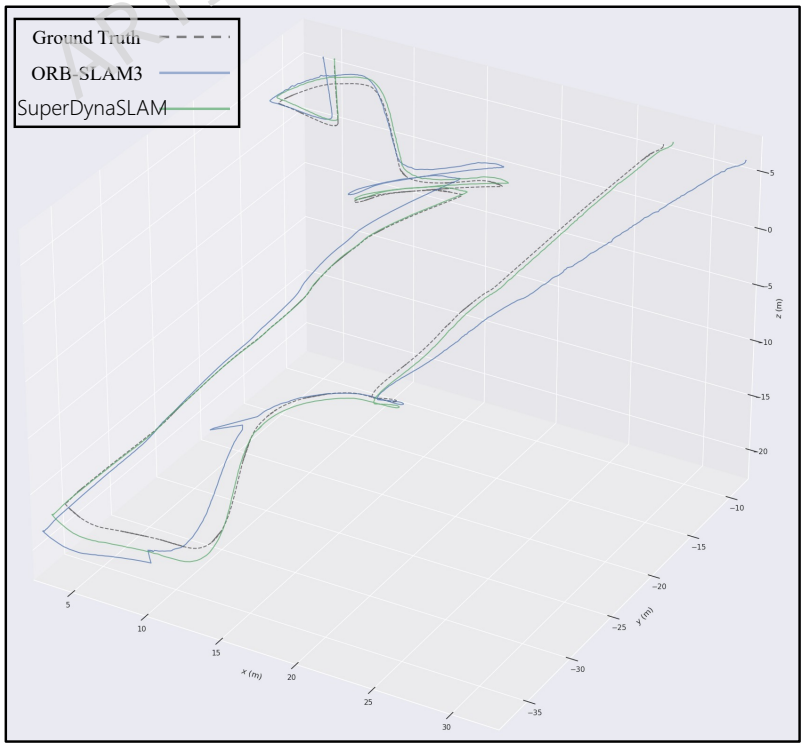


Figure 10. Trajectories of the Seq. city_day_3_high estimated by ORB-SLAM3 and ours.

Seq.	Dyna. Level	DynaSLAM		DyanVINS		ORB-SLAM3		SuperDynaSLAM	
		RPEt	RPEr	RPEt	RPEr	RPEt	RPEr	RPEt	RPEr
city_day	0_none	x	x	x	x	0.1821	**0.0604**	**0.1738**	0.0610
	1_low	x	x	x	x	0.2095	0.0599	**0.1986**	**0.0554**
	2_mid	x	x	x	x	0.1777	0.0595	**0.1775**	**0.0561**
	3_high	x	x	x	x	0.2412	0.0605	**0.2166**	**0.0553**
city_night	0_none	x	x	x	x	1.6853	0.9866	**1.5707**	**0.8766**
	1_low	x	x	x	x	3.0543	1.0658	**2.8590**	**0.9692**
	2_mid	x	x	x	x	1.9494	0.9832	**1.7233**	**0.8848**
	3_high	x	x	x	x	3.2016	0.9940	**2.7785**	**0.8635**
parking_lot	0_none	x	x	x	x	0.1221	0.0677	**0.1129**	**0.0657**
	1_low	x	x	x	x	**0.1168**	0.0675	0.1213	**0.0670**
	2_mid	x	x	x	x	0.2406	0.0915	**0.2138**	**0.0811**
	3_high	x	x	x	x	0.1200	0.0680	**0.1057**	**0.0594**

Table 6. The RPEt [m/m] and RPEr [°/m]
("x" represents the method does not provide this value in its paper)

OpenLORIS-Scene dataset evaluation

The OpenLORIS-Scene dataset is collected by using a wheeled robot in indoor environments such as market, home, and office containing several challenging factors including weak texture, dynamic objects and poor illumination. The performance was evaluated across six different sequences. Specifically, office1-1 corresponds to a static scene, office1-7 represents a low-dynamic scenario, while cafe1-1, market1-1, market1-2, and market1-3 are characterized by highly dynamic environments.

Our method is evaluated and compared with four SLAM methods including ORB-SLAM3, RDP-SLAM[37], DGO-VINS[38], and VID-SLAM[39]. All the SLAM methods are run in mono-inertial mode.

The quantitative results of AT-RMSE are summarized in Table 7 and Table 8. According to Table 7, our method achieves lower median values compared with ORB-SLAM3 in all sequences. In market1-1 sequences with many moving objects, the AT-RMSE of our method decreased the most compared to ORB-SLAM3, reaching 16.16%. This improvement is due to our method using SuperPoint and removing most of the dynamic objects in the scene. It can be found that the 95% CI widths of our method are higher than those of ORB-SLAM3 on the other sequence, except for the completely static scene of office1-1 in Table 8.

However, compared to RDP-SLAM, our method produces a larger error. This performance gap mainly arises from differences in dynamic feature handling strategies. RDP-SLAM estimates a motion probability for each individual feature point, whereas our method relies on object-level semantic masks generated by Mask R-CNN to identify potentially dynamic regions. Consequently, in sequences such as market1-1, market1-2, and market1-3, dynamic objects including shopping carts and goods that are not covered by the pre-trained Mask R-CNN categories cannot be properly identified. As a result, dynamic feature points associated with these objects are retained in the SLAM pipeline, leading to degraded localization accuracy. In contrast, the feature-level motion probability estimation adopted by RDP-SLAM enables more effective detection and removal of such dynamic features, thereby achieving higher localization accuracy in these scenarios.

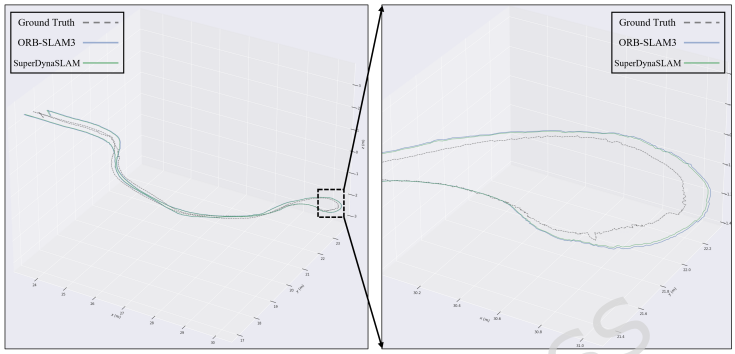
Seq.	RPD-SLAM	DGO-VINS	VID-SLAM	ORB-SLAM3	SuperDynaSLAM
office1-1	0.06378	x	0.067	0.0611	**0.060**
office1-7	**0.05899**	x	x	0.122	0.117
cafe1-1	**0.06102**	0.349	x	0.462	0.425
market1-1	**0.14088**	0.881	2.456	3.829	3.211
market1-2	**0.23461**	0.724	x	2.626	2.250
market1-3	**0.23726**	1.035	x	2.114	1.855

Table 7. AT-RMSE [m] (median)
("x" represents the method does not provide this value in its paper)

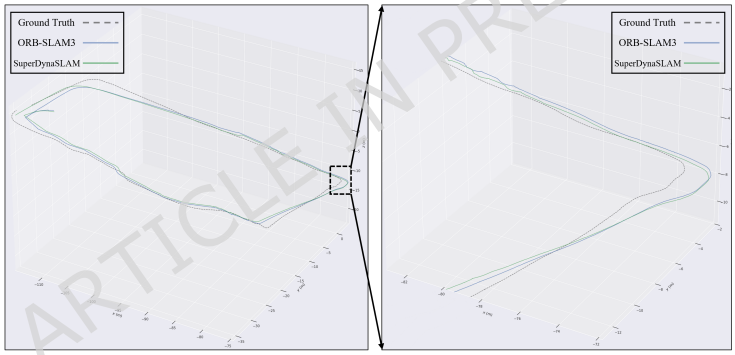
The qualitative comparison of the estimated trajectories produced by ORB-SLAM3 and the proposed method is presented in Fig. 11. As illustrated in the figure, the trajectory estimated by our method exhibits closer agreement with the ground truth than that of ORB-SLAM3, indicating improved localization accuracy.

Seq.	RPD-SLAM	DGO-VINS	VID-SLAM	ORB-SLAM3	SuperDynaSLAM
office1-1	x	x	x	**0.0611±0.0005**	0.0613±0.0005
office1-7	x	x	x	0.1227±0.0027	**0.1169±0.0026**
cafe1-1	x	x	x	0.4625±0.0222	**0.4259±0.0183**
market1-1	x	x	x	3.8304±0.4310	**3.2118±0.3347**
market1-2	x	x	x	2.6267±0.2800	**2.2512±0.1752**
market1-3	x	x	x	2.1153±0.5791	**1.8543±0.1684**

Table 8. AT-RMSE [m] (mean±95% CI)
("x" represents the method does not provide this value in its paper)



(a) Seq. cafe1-1



(b) Seq. market1-1

Figure 11. Trajectories of the Seq. cafe1-1(a) and market1-1(b) estimated by ORB-SLAM3 and ours.

The results of RPE are listed in Table 9. Our proposed method achieves lower RPEt and RPEr than ORB-SLAM3 on all sequences, but performs worse than RDP-SLAM on four sequences (cafe1-1, market1-1, market1-2, market1-3). Similar to the reason for AT-RMSE, this is because our SuperDynaSLAM cannot handle objects that the pre-trained Mask R-CNN model cannot recognize.

Ablation studies

The ablation study consists of two components. The first component assesses the contribution of each individual module to localization accuracy. The second component evaluates the robustness of the proposed method to segmentation errors, including under-segmentation and over-segmentation, and analyzes their influence on localization accuracy.

Effectiveness of individual modules

In this part, we evaluate the effectiveness of the two core modules in SuperDynaSLAM, namely the SuperPoint-based feature extraction and matching module (referred to as the S module) and the two-stage dynamic feature point detection module (referred to as the D module). To this end, we conduct ablation experiments under four different configurations: using neither module, using only the S module, using only the D module, and using both modules together. The experiments are performed

Seq.	RDP-SLAM		DGO-VINS		VID-SLAM		ORB-SLAM3		SuperDynaSLAM	
	RPEt	RPEr	RPEt	RPEr	RPEt	RPEr	RPEt	RPEr	RPEt	RPEr
office1-1	0.01189	**0.00691**	x	x	0.011	0.497	0.011	0.0121	**0.01077**	0.0123
office1-7	**0.01455**	0.01218	x	x	x	x	0.096	0.0113	0.0919	**0.0112**
cafe1-1	**0.04419**	**0.02064**	x	x	x	x	0.152	0.1071	0.13687	0.0995
market1-1	**0.05822**	**0.09071**	x	x	0.061	0.793	0.184	0.3167	0.161	0.2804
market1-2	**0.05726**	**0.08052**	x	x	x	x	0.126	0.1343	0.1138	0.1209
market1-3	**0.02806**	**0.08967**	x	x	x	x	0.284	0.1605	0.25013	0.151

Table 9. The RPEt [m/m] and RPEr [°/m]
("x" represents the method does not provide this value in its paper)

on Seq. parking_lot_3_high from the VIODE dataset and Seq. market1-1 from the OpenLORIS-Scene dataset, both of which involve highly dynamic scenes with a large number of moving objects.

Seq.	None	S module	D module	S+D module
VIODE	0.446	0.430	0.401	**0.383**
OpenLORIS-Scene	3.830	3.688	3.479	**3.211**

Table 10. AT-RMSE [m]

As shown in Table 10, the configuration without either module yields the lowest localization accuracy on both datasets. Introducing either the S module or the D module alone leads to a noticeable reduction in AT-RMSE, while the best performance is consistently achieved when both modules are jointly enabled. On the VIODE sequence, the AT-RMSE is reduced from 0.446 m to 0.383 m when both modules are used. A similar trend is observed on the OpenLORIS-Scene dataset, where the AT-RMSE decreases from 3.830 m to 3.211 m. These results indicate that the proposed method benefits from both improved feature robustness and effective dynamic feature suppression, and that the performance gains generalize across datasets with different sensing conditions and scene characteristics.

When only the S module is enabled, the SuperPoint-based front-end provides feature points with higher repeatability, more uniform spatial distribution, and more reliable correspondences, leading to improved localization accuracy compared with the baseline. However, without explicit handling of dynamic features, feature points on moving objects are still retained, which results in residual pose estimation errors in highly dynamic scenes. Conversely, when only the D module is applied, dynamic feature points can be partially suppressed using semantic and geometric constraints. Nevertheless, the reliance on hand-crafted ORB features leads to less reliable feature correspondences, which may adversely affect motion classification and limit the achievable localization accuracy. When both modules are jointly enabled, the robust feature extraction provided by SuperPoint reduces matching ambiguity, while the two-stage dynamic feature point detection method accurately identifies and removes dynamic features. This complementary behavior enables SuperDynaSLAM to achieve superior localization accuracy and robustness in dynamic environments.

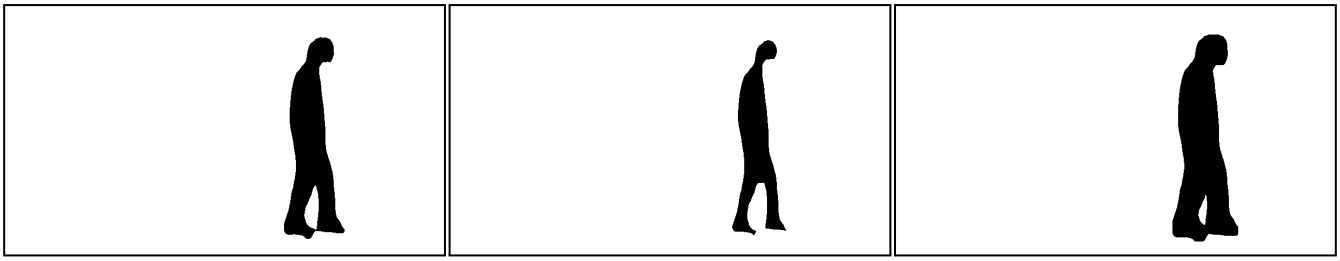
These results clearly demonstrate that both the SuperPoint-based feature extraction and matching module and the two-stage dynamic feature point detection module play important roles in enhancing the overall robustness and accuracy of SuperDynaSLAM.

Impact of segmentation errors on localization accuracy

In this part, we evaluate the robustness of the proposed method under imperfect semantic segmentation produced by Mask R-CNN, focusing on two typical error cases: under-segmentation and over-segmentation. These segmentation errors are simulated by applying morphological operations to the normal segmentation masks, where under-segmentation and over-segmentation are generated by five dilation and erosion operations, respectively, using a 3x3 kernel. Fig. 12 illustrates masks with different segmentation levels derived from the same original image. The experiments are conducted on Seq. parking_lot_3_high from the VIODE dataset and Seq. market1-1 from the OpenLORIS-Scene dataset, and the quantitative results are summarized in Table 11.

As shown in Table 11, both under-segmentation and over-segmentation negatively affect the final localization accuracy of the SLAM system, with their impact varying across datasets.

Under under-segmentation, the AT-RMSE increases by approximately 8.1-12.1% across the evaluated sequences, indicating a noticeable degradation in localization accuracy. This performance drop is mainly caused by feature points near the boundaries of dynamic objects being incorrectly classified as background features, allowing residual dynamic feature points to be introduced



(a) Normal segmentation mask **(b)** Under-segmentation **(c)** Over-segmentation

Figure 12. Semantic masks with different segmentation levels.

Seq.	Under-segmentation	Over-segmentation	Normal segmentation
VIODE	0.414	0.39	**0.383**
OpenLORIS-Scene	3.598	3.315	**3.211**

Table 11. AT-RMSE [m]

into the SLAM system and thus degrading pose estimation accuracy.

In contrast, under over-segmentation, the AT-RMSE increases by approximately 1.8–3.2%, showing a relatively smaller but still non-negligible impact on localization accuracy. Over-segmentation causes static background feature points near object boundaries to be incorrectly classified as potentially dynamic. If the object containing these misclassified features is ultimately identified as dynamic, valuable static information is discarded. Moreover, when misclassified feature points are randomly selected during motion verification, an object may be incorrectly judged as static, which, if inconsistent with its true motion state, can lead to a large number of dynamic feature points being erroneously retained, further degrading localization accuracy.

Running time analysis

For a comprehensive assessment of the proposed method, this section evaluates the running time of the major components in SuperDynaSLAM on Seq. parking_lot_3_high from the VIODE dataset and Seq. market1-1 from the OpenLORIS-Scene dataset. All experiments are conducted on a system running Ubuntu 22.04.5, equipped with an Intel Core i7-12700H CPU, 16 GB of RAM, and an NVIDIA RTX 3060 GPU with 6 GB of VRAM.

As summarized in Table 12, both SuperPoint and Mask R-CNN constitute the primary sources of computational overhead in the proposed system. Consequently, SuperDynaSLAM operates at a relatively low processing speed of approximately 2–3 FPS. In contrast, ORB-SLAM3, which serves as the baseline framework and does not incorporate either SuperPoint or Mask R-CNN, achieves a substantially higher runtime performance of approximately 25–30 FPS. Despite this increased computational cost, SuperDynaSLAM delivers improved localization accuracy compared with ORB-SLAM3, particularly in dynamic environments. This analysis highlights the computational bottleneck introduced by SuperPoint and Mask R-CNN, suggesting the need for more efficient perception components to improve runtime performance.

Seq.	SuperPoint	Mask R-CNN	Class-wise matching	Moving objects identifying
VIODE	258.39	112.28	1.13	1.59
OpenLORIS-Scene	295.68	163.51	1.14	1.74

Table 12. Running time [ms]

Failure case

This section analyzes a representative failure case to better understand the limitations of the proposed SuperDynaSLAM in complex dynamic scenes. As illustrated in Fig. 13, the scene contains a moving shopping cart that is not correctly identified as a dynamic object. Since the pre-trained Mask R-CNN model fails to detect this object, the subsequent geometric verification stage is not activated for it. As a result, feature points associated with the shopping cart are incorrectly treated as static and retained in the SLAM system.

The inclusion of these dynamic feature points introduces inconsistent geometric constraints during pose estimation, which degrades tracking accuracy and leads to increased localization error. This failure case highlights an inherent limitation of the proposed two-stage dynamic feature point detection strategy: its effectiveness depends on the ability of the semantic

(a) Original image **(b)** Failure mask

Figure 13. Failure case

segmentation module to correctly identify potentially movable objects in the scene. When dynamic objects fall outside the predefined categories of the segmentation model, they may bypass the dynamic feature filtering process entirely.

It is worth noting that this limitation does not stem from the geometric verification stage itself, but rather from the restricted category coverage of the pre-trained segmentation network. In scenarios involving uncommon or previously unseen dynamic objects, such as shopping carts or transported goods, the current system may therefore fail to suppress all dynamic feature points.

This analysis underscores the importance of improving object-level perception for dynamic SLAM systems and motivates the investigation of more generalizable object recognition strategies. A broader discussion of potential extensions and future research directions is provided in Section "Conclusions and future work".

Conclusions and future work

To enhance the accuracy of SLAM in dynamic environments, we propose SuperDynaSLAM which integrates a SuperPoint-based feature extraction and matching method with a two-stage dynamic feature point detection method fusing semantic information and geometric constraints. The replacement of hand-crafted ORB extractor with SuperPoint enhances the robustness and spatial distribution of the feature points under challenging conditions. By constructing the geometric constraints from the IMU measurements to identify moving objects within semantic masks, SuperDynaSLAM accurately removes dynamic features that could degrade pose estimation.

To comprehensively evaluate the proposed SuperDynaSLAM, extensive experiments were conducted on multiple public datasets, including EuRoC, VIODE, and OpenLORIS-Scene. The results demonstrate that SuperDynaSLAM consistently achieves improved localization accuracy over ORB-SLAM3 and other state-of-the-art methods, particularly in dynamic scenes. Moreover, the ablation studies and sensitivity analyses confirm that both the SuperPoint-based feature extraction and matching method and the two-stage dynamic feature point detection method play complementary and essential roles in the overall performance gains.

Despite these encouraging results, several challenges remain for future investigation. First, to overcome the limited category coverage of the current segmentation model, future work will explore more comprehensive segmentation approaches, such as the Segment Anything Model, to improve generalization to previously unseen dynamic objects. Second, motivated by the runtime analysis, we will investigate more lightweight feature extraction and semantic perception networks to reduce computational overhead and enhance real-time performance. Third, we plan to extend SuperDynaSLAM toward a multi-sensor framework by incorporating LiDAR, with the aim of further improving robustness and reliability in complex dynamic environments.

Data availability

The datasets analysed during the current study are available in the KITTI dataset, EuRoC dataset, VIODE dataset and OpenLORIS-Scene dataset repositories, at https://www.cvlibs.net/datasets/kitti/eval_odometry.php, https://projects.asl.ethz.ch/datasets/euroc-mav/, https://github.com/kminoda/VIODE, and https://lifelong-robotic-vision.github.io/dataset/scene.html.

Code availability

The code for the proposed SuperDynaSLAM is available at https://doi.org/10.5281/zenodo.19142293.

References

1. Ebadi, K. *et al.* Present and future of slam in extreme environments: The darpa subt challenge. *IEEE Transactions on Robotics* **40**, 936–959 (2023).
2. Alsadik, B. & Karam, S. The simultaneous localization and mapping (slam): An overview. *Surv. geospatial engineering journal* **1**, 1–12 (2021).
3. Yuan, S., Wang, H. & Xie, L. Survey on localization systems and algorithms for unmanned systems. *Unmanned Syst.* **9**, 129–163 (2021).
4. Mur-Artal, R., Montiel, J. M. M. & Tardos, J. D. Orb-slam: A versatile and accurate monocular slam system. *IEEE transactions on robotics* **31**, 1147–1163 (2015).
5. Rublee, E., Rabaud, V., Konolige, K. & Bradski, G. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, 2564–2571 (Ieee, 2011).
6. Vidal, A. R., Rebecq, H., Horstschaefer, T. & Scaramuzza, D. Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high-speed scenarios. *IEEE Robotics Autom. Lett.* **3**, 994–1001 (2018).
7. Gonzalez, R. C. *Digital image processing* (Pearson education india, 2009).
8. Campos, C., Elvira, R., Rodríguez, J. J. G., Montiel, J. M. & Tardós, J. D. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE transactions on robotics* **37**, 1874–1890 (2021).
9. Derpanis, K. G. Overview of the ransac algorithm. *Image Rochester NY* **4**, 2–3 (2010).
10. Wang, Y., Tian, Y., Chen, J., Xu, K. & Ding, X. A survey of visual slam in dynamic environment: The evolution from geometric to semantic approaches. *IEEE Transactions on Instrumentation Meas.* **73**, 1–21 (2024).
11. DeTone, D., Malisiewicz, T. & Rabinovich, A. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 224–236 (2018).
12. He, K., Gkioxari, G., Dollár, P. & Girshick, R. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2961–2969 (2017).
13. Hartley, R. & Zisserman, A. *Multiple view geometry in computer vision* (Cambridge university press, 2003).
14. Jaimez, M., Kerl, C., Gonzalez-Jimenez, J. & Cremers, D. Fast odometry and scene flow from rgb-d cameras based on geometric clustering. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 3992–3999 (IEEE, 2017).
15. Song, B. *et al.* Dgm-vins: Visual–inertial slam for complex dynamic environments with joint geometry feature extraction and multiple object tracking. *IEEE Transactions on Instrumentation Meas.* **72**, 1–11 (2023).
16. Dai, W., Zhang, Y., Li, P., Fang, Z. & Scherer, S. Rgb-d slam in dynamic environments using point correlations. *IEEE transactions on pattern analysis machine intelligence* **44**, 373–389 (2020).
17. Kim, D.-H. & Kim, J.-H. Effective background model-based rgb-d dense visual odometry in a dynamic environment. *IEEE Transactions on Robotics* **32**, 1565–1573 (2016).
18. Lin, H.-Y., Liu, T.-A. & Lin, W.-Y. Inertialnet: Inertial measurement learning for simultaneous localization and mapping. *Sensors* **23**, 9812 (2023).
19. Lai, D., Zhang, Y. & Li, C. A survey of deep learning application in dynamic visual slam. In *2020 International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE)*, 279–283 (IEEE, 2020).
20. Zhong, F., Wang, S., Zhang, Z. & Wang, Y. Detect-slam: Making object detection and slam mutually beneficial. In *2018 IEEE winter conference on applications of computer vision (WACV)*, 1001–1010 (IEEE, 2018).
21. Hu, X. *et al.* Cfp-slam: A real-time visual slam based on coarse-to-fine probability in dynamic environments. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4399–4406 (IEEE, 2022).
22. Bescos, B., Fácil, J. M., Civera, J. & Neira, J. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE robotics automation letters* **3**, 4076–4083 (2018).
23. Vincent, J. *et al.* Dynamic object tracking and masking for visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4974–4979 (IEEE, 2020).
24. Bolya, D., Zhou, C., Xiao, F. & Lee, Y. J. Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9157–9166 (2019).

25. Yan, L. *et al.* Dgs-slam: A fast and robust rgbd slam in dynamic environments combined by geometric and semantic information. *Remote. Sens.* **14**, 795 (2022).

26. 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 Transactions on Instrumentation Meas.* **72**, 1–12 (2022).

27. Mur-Artal, R. & Tardós, J. D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics* **33**, 1255–1262 (2017).

28. Qin, L. *et al.* Rso-slam: A robust semantic visual slam with optical flow in complex dynamic environments. *IEEE Transactions on Intell. Transp. Syst.* **25**, 14669–14684 (2024).

29. Geiger, A., Lenz, P. & Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2012).

30. Burri, M. *et al.* The euroc micro aerial vehicle datasets. *The Int. J. Robotics Res.* **35**, 1157–1163 (2016).

31. Minoda, K., Schilling, F., Wüest, V., Floreano, D. & Yairi, T. Viode: A simulated dataset to address the challenges of visual-inertial odometry in dynamic environments. *IEEE Robotics Autom. Lett.* **6**, 1343–1350 (2021).

32. Shi, X. *et al.* Are we ready for service robots? the OpenLORIS-Scene datasets for lifelong SLAM. In *2020 International Conference on Robotics and Automation (ICRA)*, 3139–3145 (2020).

33. Sturm, J., Engelhard, N., Endres, F., Burgard, W. & Cremers, D. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, 573–580 (IEEE, 2012).

34. Ge, Y., Zhang, L., Wu, Y. & Hu, D. Pipo-slam: Lightweight visual-inertial slam with preintegration merging theory and pose-only descriptions of multiple view geometry. *IEEE Transactions on Robotics* **40**, 2046–2059 (2024).

35. Qin, T., Cao, S., Pan, J. & Shen, S. A general optimization-based framework for global pose estimation with multiple sensors. *arXiv preprint arXiv:1901.03642* (2019).

36. Song, S., Lim, H., Lee, A. J. & Myung, H. Dynavins: A visual-inertial slam for dynamic environments. *IEEE Robotics Autom. Lett.* **7**, 11523–11530 (2022).

37. Zhang, H., Wang, D. & Huo, J. Real-time dynamic slam using moving probability based on imu and segmentation. *IEEE Sensors J.* **24**, 10878–10891 (2024).

38. Liu, X., Zhang, Y., Lu, G., Li, S. & Liu, J. Dgo-vins: A visual-inertial slam for dynamic environments with geometric constraint and adaptive state optimization. *IEEE Robotics Autom. Lett.* (2025).

39. Zhang, C., Gu, S., Li, X., Deng, J. & Jin, S. Vid-slam: A new visual inertial slam algorithm coupling an rgb-d camera and imu based on adaptive point and line features. *IEEE Sensors J.* (2024).

Acknowledgements

This work was supported by the National Nature Science Foundation of China [grant number 61374197]; the Shanghai Nature Science Foundation of Shanghai Science and Technology Commission [grant number 20ZR1437900]; the Jiangsu Basic Science Foundation of Colleagues and Universities (nature science) [grant number 24KJD520009].

Author contributions statement

Conceptualization, J.C. and Y.H.; methodology, J.C. and Y.H.; software, J.C.; validation, J.C., Y.H. and L.W.; formal analysis, J.C. and Y.H.; investigation, J.C.; writing—original draft preparation, J.C. and Y.H.; writing—review and editing, J.C., Y.H. and L.W.; project administration, Y.H. and L.W.; funding acquisition, Y.H. and L.W.. All authors have read and agreed to the published version of the manuscript.

Competing interests

The authors declare no competing interest.