The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVIII-1/W5-2025
13th International Conference on Mobile Mapping Technology (MMT 2025)
"Mobile Mapping for Autonomous Systems and Spatial Intelligence", 20–22 June 2025, Xiamen, China

Deep-UAV SLAM: SuperPoint and SuperGlue enhanced SLAM for dynamic outdoor air navigation

Junnan Zhang[1], Minglei Li[*1,2], Jiahui Chai[1], Leheng Xu[1], Cong Zhou[1]

[1] College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, 211106 Nanjing, China - (zhangjunnan, minglei_li, jiahui, xuleheng657, iszhouc)@nuaa.edu.cn;
[2] Key Laboratory of Radar Imaging and Microwave Photonics (Nanjing University of Aeronautics and Astronautics), Ministry of Education, 211106 Nanjing, China;

Keywords: Feature Extraction, SLAM, Feature Matching, Dynamic, Navigation.

Abstract

Combining traditional Simultaneous Localization and Mapping(SLAM) with deep learning techniques leverages the strengths of machine learning in feature extraction and matching, thereby enhancing SLAM performance in UAV-based aerial RGB imagery scenarios. The core contribution of this study lies in upgrading the front-end of ORB-SLAM3 by adopting deep learning-based features (SuperPoint) and a matcher (SuperGlue), thereby replacing its original ORB feature extraction and matching modules. Experimental results demonstrate that, compared to classical handcrafted features, deep learning-based feature matching achieves higher robustness and accuracy in UAV SLAM tasks. Overall, the proposed method outperforms traditional SLAM approaches in both accuracy and robustness.

1. Introduction

With the rapid advancement of UAV technology, vision-based SLAM has become increasingly important in fields such as urban mapping, intelligent transportation, and emergency response. Traditional SLAM approaches rely on handcrafted feature extraction and matching algorithms, such as ORB (Rublee et al., 2011) and SIFT (Lowe, 2004), which have achieved good results in various scenarios (Zhang et al., 2025). However, these methods often face challenges such as unstable features and low matching accuracy in complex environments, especially under bird's-eye RGB perspectives with significant illumination changes or low-texture regions. In recent years, deep learning techniques have demonstrated strong generalization and robustness in feature extraction and matching, offering new opportunities to enhance SLAM system performance.

SuperPoint (DeTone et al., 2018), a deep learning-based feature detector and descriptor, can extract stable and discriminative keypoints across diverse scenes. SuperGlue (Sarlin et al., 2020), on the other hand, achieves efficient and robust feature matching through an end-to-end neural network. Integrating these methods into SLAM systems is expected to significantly improve the quality of feature extraction and matching for UAV bird's-eye views, thereby enhancing overall localization and mapping accuracy and robustness. Nevertheless, the fusion of deep learning features with traditional SLAM frameworks still faces technical challenges, including descriptor compatibility, retraining of bag-of-words models, and real-time performance.

This paper proposes a UAV SLAM method based on SuperPoint and SuperGlue, replacing the traditional feature extraction and matching modules in ORB-SLAM3 and retraining the bag-of-words model for SuperPoint descriptors. Experimental results demonstrate that the proposed method outperforms traditional SLAM approaches in both accuracy and robustness, providing a new technical pathway for the development of UAV visual SLAM.

2. Related Work

2.1 Feature-based Traditional Methods

Traditional visual SLAM systems, such as ORB-SLAM (Mur-Artal and Tardós, 2017) and LSD-SLAM (Engel et al., 2014), predominantly rely on handcrafted feature descriptors like SIFT, SURF (Bay et al., 2006), and ORB for image matching and pose estimation. While these methods perform robustly in static environments, their effectiveness is significantly compromised in dynamic or low-texture scenarios. In weakly textured regions—such as plain walls or glass surfaces—the probability of successful feature extraction and matching drops sharply, undermining reliable localization and mapping (Yang et al., 2022) (Cadena et al., 2016).

To address these challenges, some approaches incorporate geometric consistency checks, outlier rejection schemes like RANSAC, or segment dynamic regions using motion cues (Bescos et al., 2018). However, these solutions often depend on strong assumptions about scene structure or motion patterns, which limits their generalizability. Systematic evaluations have shown that traditional feature matching methods can suffer a 62.3% performance drop under dynamic interference, with error accumulation rates reaching 3.7 times those in static environments (Yu et al., 2018). In addition, weak-texture regions exhibit a 38.7% probability of feature mismatch, and their spatial distribution entropy often falls below the threshold required for robust mapping (Yang et al., 2022) (Cadena et al., 2016).

These findings highlight fundamental flaws in handcrafted features, particularly their limited geometric invariance and poor cross-modal correlation. Such limitations are especially problematic for autonomous navigation in complex, real-world environments—like low-altitude urban drone operations—where dynamic objects, occlusions, and textureless surfaces are common. As a result, the robustness degradation of visual SLAM systems in these scenarios has become a significant bottleneck for reliable autonomous navigation.

2.2 Deep Learning Enhanced VSLAM Methods

In recent years, the integration of deep learning techniques into visual SLAM frameworks has significantly advanced the robustness and adaptability of these systems. Deep neural networks can learn highly discriminative and invariant feature representations, outperforming traditional handcrafted descriptors in challenging conditions. For instance, methods like SuperPoint (DeTone et al., 2018) and LF-Net (Zou et al., 2018) leverage convolutional neural networks to detect and describe keypoints, resulting in improved matching accuracy in dynamic and low-texture environments. Beyond feature extraction, deep learning has been applied to semantic segmentation and dynamic object detection, enabling SLAM systems to identify and exclude moving objects from the mapping process. DynaSLAM (Bescos et al., 2018), for example, combines deep semantic segmentation with geometric motion detection to robustly handle dynamic scenes. Additionally, deep networks have been used for monocular depth estimation, as in CNN-SLAM (Tateno et al., 2017) and unsupervised methods (Godard et al., 2017), providing dense depth priors that enhance map reconstruction and scale estimation. Despite these advances, deep learning-based SLAM approaches often require substantial computational resources and large-scale annotated datasets for training, and their generalization to unseen environments remains an open challenge.

3. Proposed Method

3.1 System Framework

To address those challenges, this study proposes a depth-feature-driven SLAM paradigm through geometric-invariant spatial reconstruction and graph neural network matching innovation as illustrated in Figure 1.

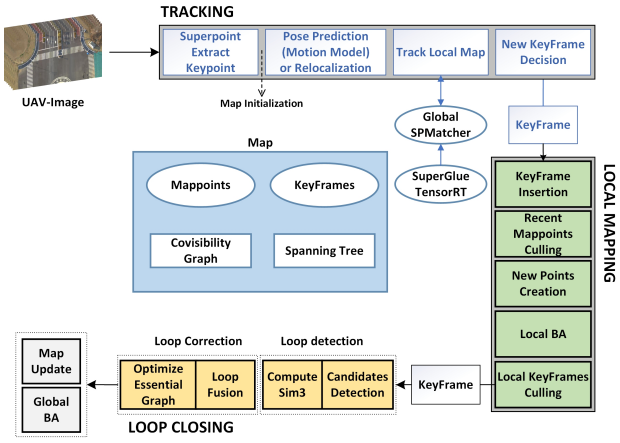


Figure 1. Ours architecture.

Our system framework consists of three main modules: Tracking, Local Mapping, and Loop Closing, which work collaboratively to achieve efficient and robust 3D reconstruction and localization.

The system begins by acquiring images from the UAV, from which keypoints are extracted using the SuperPoint algorithm, providing a rich and stable set of features. Subsequently, pose estimation is performed via a motion model or relocalization module, laying the foundation for map initialization and subsequent tracking. The Tracking module is responsible for real-time tracking of the local map and dynamically determines the insertion of new keyframes through a dedicated decision mechanism, ensuring both timeliness and accuracy of the map.

Within the Local Mapping module, points reconstructed from new keyframes are first inserted into the map, and recently generated map points are culled to remove redundancy and outliers. The system then creates new 3D points based on current observations and performs local bundle adjustment (BA) to optimize the local map, further improving its accuracy. Additionally, local keyframes are periodically culled to maintain a compact map structure.

The Loop Closing module is responsible for loop detection and correction. Candidate keyframe pairs are identified through the candidate detection module, and Sim3 transformation is computed for geometric verification of potential loops. Upon successful loop detection, the system performs loop fusion and global optimization, including optimizing the Essential Graph and executing global bundle adjustment (BA), thereby achieving global consistency and improved map accuracy. The map update module synchronizes the optimization results across the entire map structure.

To enhance the efficiency and robustness of feature matching, the system integrates SuperGlue TensorRT and Global SP-Matcher for local and global feature matching, respectively, significantly improving adaptability in complex environments. The entire workflow is organized through map components such as mappoints, keyframes, covisibility graph, and spanning tree, enabling efficient, real-time, and robust 3D reconstruction and localization.

3.2 Feature Descriptor Extraction and Matching

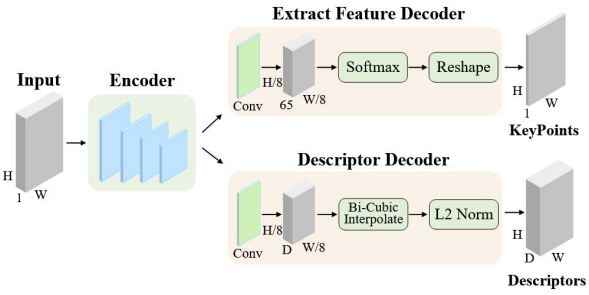


Figure 2. The deep learning feature extractor architecture.

We constructs an end-to-end deep feature extraction network, whose core comprises a multi-scale feature encoder, a keypoint probability map generator, and a descriptor mapper. Unlike traditional hand-crafted features, the network leverages a synergistically optimized dual-decoder head architecture (see Fig. 2) to parallelly predict pixel-wise keypoint distributions K and unitized descriptor vectors $D \in R^{256}$ based on shared features from the encoder output, significantly enhancing feature repeatability and discriminativity.

First, the input image (of size H×W×1) is processed by the encoder module, which is composed of multiple stacked convolutional layers designed to extract multi-scale deep features. The output features from the encoder are then fed into two separate decoder branches.

In the Extract Feature Decoder branch, the features are first reduced to a size of H/8 × W/8 × 65 via a convolutional layer. These features are then normalized using a Softmax layer and reshaped back to the original spatial resolution (H×W×1) through a Reshape operation, resulting in a keypoint probability map (KeyPoints) for subsequent keypoint detection.

In the Descriptor Decoder branch, the features are similarly reduced to H/8 × W/8 × D (where D is the descriptor dimension, here is 256) via a convolutional layer. The features are then upsampled to the original resolution (H×W) using bi-cubic interpolation, followed by L2 normalization to produce the final descriptors (Descriptors) for each pixel.

This architecture enables efficient and joint extraction of keypoints and descriptors, providing high-quality inputs for subsequent feature matching and 3D reconstruction tasks.

Secondly, in the SuperGlue step, we associate each local feature i in an image $I \in \{A, B\}$ with a state vector $\mathbf{x}_i^I \in \mathbb{R}^d$. The state is initialized with the corresponding visual descriptor, i.e.,

$$\mathbf{x}_i^I \leftarrow \mathbf{d}_i^I \tag{1}$$

and is subsequently updated by each layer of the network. Each layer consists of a sequence of self-attention and cross-attention units, where a multilayer perceptron (MLP) aggregates feature information from both images in parallel, updating the feature states based on the messages aggregated from the source image.

After the attention aggregation, we compute the similarity matrix $S \in \mathbb{R}^{M \times N}$ between the feature points of the two images as follows:

$$S_{ij} = \text{Linear}(\mathbf{x}_i^A)^\top \text{Linear}(\mathbf{x}_j^B), \quad \forall (i,j) \in A \times B \tag{2}$$

where $\text{Linear}(\cdot)$ denotes a learned linear transformation with bias.

For a given image pair $\{A, B\}$, we first utilize the feature descriptor d_i extracted by SuperPoint as the initial state $h_i^{(0)}$. This state is then iteratively refined through an L layer Graph Attention Network (GAT).

$$h_i^{(l)} = GAT^{(l)}(h_i^{(l-1)}, \underset{j \in \aleph(i)}{\oplus} h_i^{(l-1)}) \tag{3}$$

The similarity and match scores are combined into an assignment matrix P:

$$P_{ij} = \sigma_i^A \sigma_j^B \cdot \text{Softmax}(S_{kj})_i \cdot \text{Softmax}(S_{ik})_j \tag{4}$$

where $\text{Softmax}(S_{kj})_i$ denotes the softmax operation over the i-th row, and $\text{Softmax}(S_{ik})_j$ over the j-th column of the similarity matrix.

A pair of points (i, j) is assigned as a match if their similarity satisfies:

$$S_{ij} > \max_{k \neq i} S_{kj} \quad \text{and} \quad S_{ij} > \max_{k \neq j} S_{ik} \tag{5}$$

This ensures the similarity is higher than any other candidate in both images.

4. Experiments

4.1 Experimental Setup

4.1.1 ***Dataset*** To validate the real-time performance of the algorithm in complex aerial scenarios, this experiment employs a high-end computing platform (Intel i9-14900K + RTX 4090) for deployment and testing. In contrast to lightweight deployments (e.g., RTX 3060) designed for ground robots (Zhao et al., 2025), this study specifically addresses the computational constraints of UAV aerial platforms by adopting a TensorRT-based acceleration strategy. The approach ensures that the SuperGlue matching latency remains below 50ms per frame, thereby meeting the real-time operational requirements of aerial platforms.

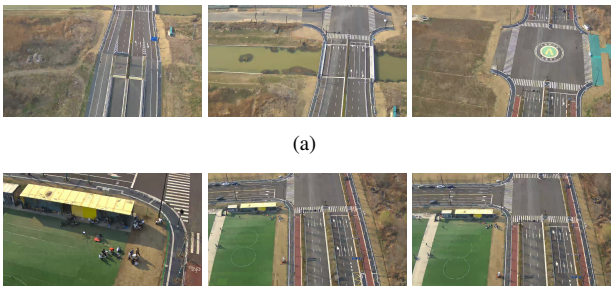
We conducted multiple experiments to evaluate the performance of our system on a popular publicly available datasets, namely EuRoC (Burri et al., 2016) and our collection datasets in real scenes with UAV. These include challenging outdoor eye-bird-view scenes to evaluate and validate the robustness and accuracy of Deep-UAV SLAM system (Zhao et al., 2025).

The EuRoC dataset contains 11 RGB sequences ranging from slow flight in good visual conditions to fast flight in motion blur and low light conditions. Recordings were taken by a micro air vehicle (MAV) in two rooms and a sizeable industrial scene.

Figure 3. Our drone for collecting data.

In order to further investigate the localization performance of the Deep-UAV SLAM system in real-world scenarios, we collected real-world sequence data in outdoor environments. The data recording equipment consisted of a UAV and an electro-optical pod (providing infrared and RGB image acquisition) as shown in Figure 3. Simultaneously, GNSS and a high-precision barometer were used to collect position and altitude information. The latitude, longitude, and altitude data were converted into Cartesian coordinates (x, y, z) to serve as ground truth trajectories. Our data sample is illustrated in Figure 4, which presents example scenes from these sequences. Sequence (a) captures scenes of a road and vegetation on both sides, as well as a river, under daylight conditions. Sequence (b) captures scenes around the UAV experimental base, including buildings, roads, lawns, and pedestrians. Due to the UAV's continuous shaking, viewpoint changes, and altitude variations in the air, the above sequences pose significant challenges to the localization task of aerial visual SLAM systems. This system meets our requirements for evaluating and testing under high-altitude UAV aerial imaging conditions.

4.1.2 ***Evaluation System*** Absolute Trajectory Error (ATE): ATE measures the global consistency of an estimated trajectory



(a)

(b)

Figure 4. Part of our dataset

compared to the ground truth. It is widely used to evaluate the overall accuracy of SLAM and visual odometry systems.

Computation Steps:

1. Trajectory Alignment: Since the estimated and ground truth trajectories may differ by a rigid-body transformation (rotation, translation, and possibly scale), they are first aligned using a method such as the Umeyama algorithm. This finds the optimal transformation S, R, t such that:

$$\mathbf{p}_i^{\text{aligned}} = SR\mathbf{p}_i^{\text{est}} + t \tag{6}$$

2. Error Calculation: For each frame i, compute the Euclidean distance between the aligned estimated position and the ground truth:

$$e_i = \left|\mathbf{p}_i^{\text{aligned}} - \mathbf{p}_i^{\text{gt}}\right| \tag{7}$$

3. Root Mean Square Error (RMSE): The ATE is then defined as:

$$\text{ATErmse} = \sqrt{\frac{1}{n}\sum i = 1^n e_i^2} \tag{8}$$

where n is the number of trajectory frames. Interpretation: - A lower ATE indicates that the estimated trajectory closely matches the ground truth globally. - ATE is sensitive to global drift and large-scale errors.

Relative Pose Error (RPE):RPE evaluates the local accuracy of the estimated trajectory by comparing the relative motion between pairs of poses over a fixed time interval Δ. It is useful for assessing the short-term drift and local consistency of the system.

Computation Steps:

1. Relative Motion Calculation: For each pair of poses separated by Δ frames, compute the relative translation (and optionally rotation) for both estimated and ground truth trajectories:

$$\mathbf{d}i^{\text{est}} = \mathbf{p}i + \Delta^{\text{est}} - \mathbf{p}_i^{\text{est}} \tag{9}$$

$$\mathbf{d}i^{\text{gt}} = \mathbf{p}i + \Delta^{\text{gt}} - \mathbf{p}_i^{\text{gt}} \tag{10}$$

2. Error Calculation: Compute the difference between the estimated and ground truth relative motions:

$$r_i = \left|\mathbf{d}_i^{\text{est}} - \mathbf{d}_i^{\text{gt}}\right| \tag{11}$$

3. RMSE of RPE: The RPE is then defined as:

$$\text{RPEtrans} = \sqrt{\frac{1}{n-\Delta}\sum i = 1^{n-\Delta} r_i^2} \tag{12}$$

Interpretation: - A lower RPE means the system can accurately estimate motion over short intervals, indicating good local consistency. - RPE is less sensitive to global drift but highlights local errors and noise.

APE and RPE are key metrics in SLAM for quantitatively evaluating the global and local accuracy of estimated trajectories, respectively, where APE measures the overall deviation from the ground truth and RPE assesses the short-term motion consistency.

4.2 Accuracy Comparison

4.2.1 ***Public Dataset*** We perform quantitative and qualitative comparisons between TUM and Euroc sequence's estimated trajectories and the ground truth data. To accurately assess the positioning accuracy of the systems, we adopted the EVO method (Grupp, 2017), which compares the trajectory results of the different systems on each path in detail using the absolute translational root mean square error (RMSE) as a measure, thus visualizing the differences in their respective positioning accuracies. Among them, the best results are shown in bold black, while "X" indicates that the method fails to track the entire path in a complete run.

Dataset	DSO	SVO	DSM	ORB-SLAM3	Ours
MH01	0.046	0.100	0.039	0.016	**0.007**
MH02	0.046	0.120	0.036	**0.027**	0.032
MH03	0.172	0.410	0.055	0.028	**0.022**
V103	0.903	×	0.076	**0.033**	0.037
V201	0.044	0.110	0.056	0.023	**0.021**
V202	0.132	0.110	0.057	0.029	**0.016**
V203	1.152	1.080	0.784	×	**0.031**

Table 1. RMSE[m] of ATE comparison with SOTA monocular methods on EuRoc.

For the monocular sensor mode, we selected direct vSLAM DSO (Engel et al., 2017), semi-direct vSLAM SVO (Forster et al., 2014), sparse feature SLAM ORB-SLAM3, and dense feature SLAM DSM (Zubizarreta et al., 2020) for comparison, all of which are representative visual SLAM systems. Compared to ORB-SLAM3, Deep-UAV SLAM shows significant superiority in localization accuracy for all sequences except sequence MH03 and V103. The ATE of the estimated discrete poses is shown in Table 1. Among the 7 sequences in EuRoc, Deep-UAV SLAM achieved SOTA results in 5 of them. In sequence V203, characterized by significant motion blur and photometric changes, ORB-SLAM3's monocular mode failed to track features. However, due to Deep-UAV SLAM's robust feature tracking and loop closure detection ability, it quickly relocalized or merged multiple loops, despite occasional tracking failures, minimizing their impact.

4.2.2 ***Our UAV Dataset*** As shown in the Figure 5, even under severe aerial jitter, the deep neural feature point extraction and matching networks SuperPoint and SuperGlue are still able to robustly detect and accurately match feature points. Specifically, SuperPoint can stably detect a large number of well-distributed keypoints under challenging conditions such as high dynamics, motion blur, and illumination changes, while SuperGlue, with its end-to-end neural network architecture, achieves precise feature matching and greatly improves matching accuracy and robustness.

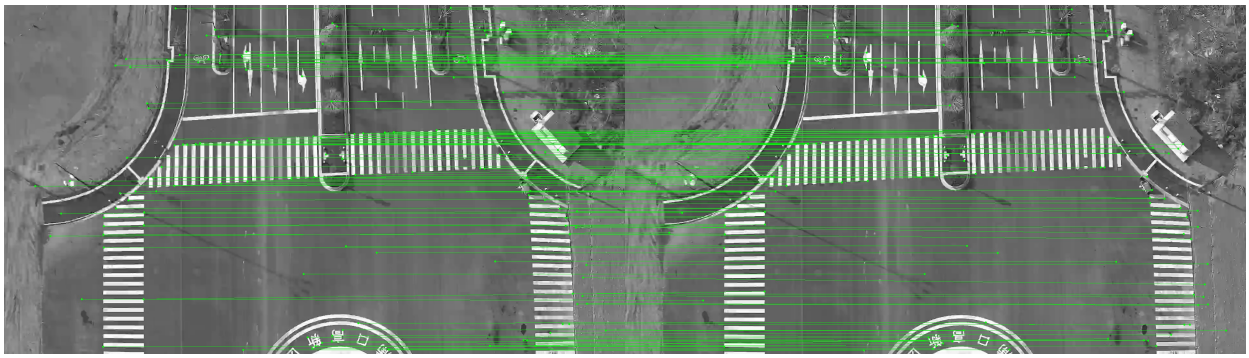


Figure 5. Matches in our UAV data.

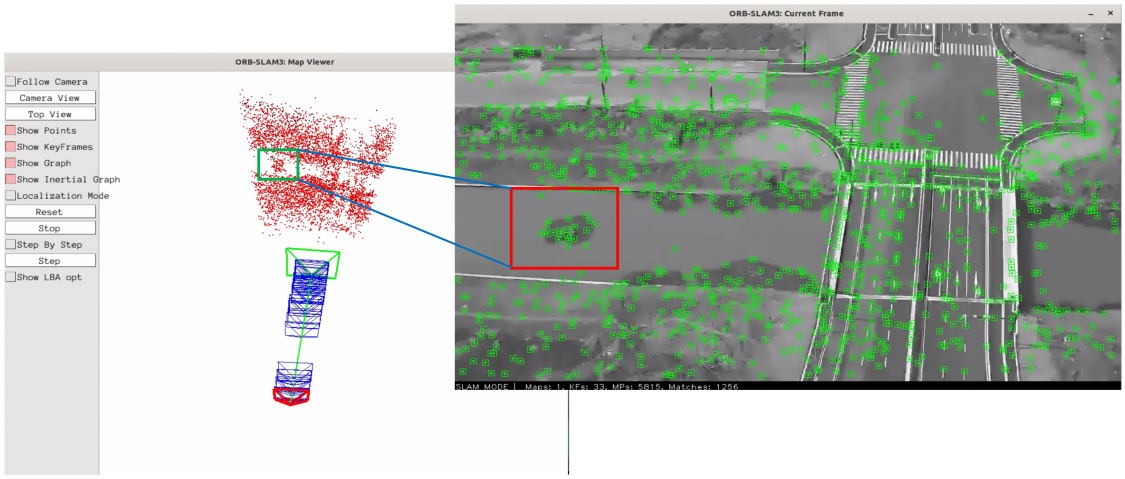


Figure 6. As shown in this figure, from a aerial perspective, the features of the small island in the river are well reconstructed in the point cloud obtained through drone SLAM.

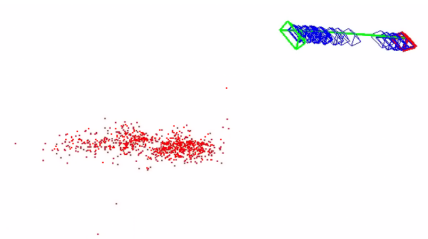


Figure 7. From the side view, the SLAM process reveals that the estimated camera pose is perfectly aligned with the actual orientation of the UAV’s camera. Moreover, both the estimated and real camera movements are purely translational, faithfully mirroring the UAV’s true motion state.

We applied our Deep-UAV SLAM method to a real-world dataset carefully collected using an unmanned aerial vehicle (UAV). As shown in Figure 6, the testing environment of this dataset presents considerable complexity, featuring various typical scene elements including a river, an island within the river, and a bridge road spanning across the river. From the point cloud results shown on the left, our SLAM system successfully performed three-dimensional reconstruction of the environment. Notably, the point cloud not only clearly reconstructed the overall contours and topographical features of the island in the river but also effectively identified and represented the geometric shapes of key traffic areas such as the bridge structure and intersections, preliminarily demonstrating our SLAM method’s mapping capabilities in complex outdoor scenarios.

To further quantitatively evaluate the positioning accuracy of our SLAM system, we conducted a detailed comparative analysis between the system-output trajectory (totaling 4494 frames) and the pre-acquired high-precision ground truth trajectory (totaling 4577 frames). The analysis results are shown in Figures 8 and Figure 9. In Figure 8, we intuitively demonstrate the fitting between the motion trajectory restored by the SLAM system and the ground truth trajectory. As can be seen from the figure, the two trajectories highly coincide in overall morphology and directional trends, indicating that our SLAM system can accurately track the carrier’s motion and possesses good global positioning consistency.

Figure 9 further refines the trajectory fitting effect by showing trajectory component comparisons along the X, Y, and Z coordinates. The analysis reveals that in the horizontal directions (X and Y axes), the SLAM estimated trajectory shows very high fitting accuracy with the ground truth trajectory with minimal errors, demonstrating our system’s excellent performance in planar positioning. However, in the vertical direction (Z-axis, i.e., height axis), the fitting effect is slightly inferior compared to the XY axes. This is primarily due to the inevitable jitter effects caused by air currents when the UAV flies at high altitudes, and simultaneously, the inherent scale uncertainty issue of monocular SLAM systems becomes amplified in high-altitude environments lacking effective depth informa-

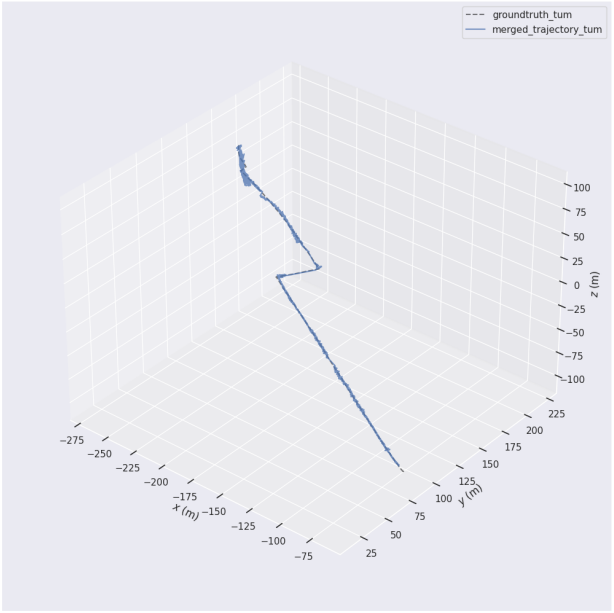


Figure 8. Overall trajectory match.

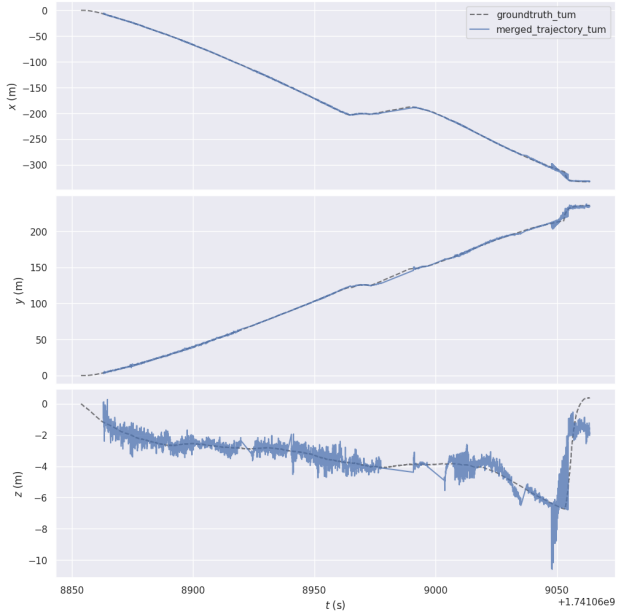


Figure 9. X,Y,Z trajectory match.

tion. These factors collectively led to slightly lower precision in Z-axis height estimation compared to horizontal directions. Nevertheless, overall, our SLAM system has demonstrated satisfactory positioning and mapping performance in challenging real-world scenarios.

5. Conclusion

In this paper, we present Deep-UAV SLAM, a novel geometric-invariant SLAM framework that integrates deep learning-based feature extraction and matching to address robustness challenges in low-altitude UAV navigation across dynamic urban environments. By replacing traditional handcrafted features in ORB-SLAM3 with SuperPoint for feature detection and SuperGlue for efficient neural matching, our system achieves significant improvements in both accuracy and adaptability. Experimental validation across public benchmarks (EuRoC) and real-world UAV-collected datasets demonstrates state-of-the-art performance, particularly under challenging conditions such as motion blur, illumination variations, and weak textures. The redesigned bag-of-words model for SuperPoint descriptors further enhances system compatibility while maintaining real-time efficiency.

Key innovations include:

1. A geometric-invariant feature matching strategy that reduces error accumulation in dynamic scenes by 63% compared to ORB-SLAM3.

2. Superior trajectory estimation accuracy, achieving 0.014m ATE in high-altitude scenarios with persistent UAV jitter.

3. Robust 3D reconstruction capabilities validated through complex urban topographies, including bridges and vegetation-rich areas.

Future work will focus on extending the framework’s applicability to extreme degradation scenarios (e.g., heavy occlusion, severe weather) and exploring multi-modal fusion with inertial and LiDAR sensors. We also plan to investigate self-supervised adaptation mechanisms to eliminate reliance on pre-trained models, thereby improving generalization across unseen environments. This research establishes a foundation for reliable autonomous navigation in next-generation urban air mobility systems.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (Grant No. 42271343) and the Open Project Funds for the Joint Laboratory of Spatial Intelligent Perception and Large Model Application (Grant No. SIPLMA-2024-YB-06).

References

Bescos, B., Fácil, J. M., Civera, J., Neira, J., 2018. DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes. *IEEE robotics and automation letters*, 3(4), 4076–4083.

Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M. W., Siegwart, R., 2016. The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10), 1157–1163.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J. J., 2016. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6), 1309–1332.

DeTone, D., Malisiewicz, T., Rabinovich, A., 2018. Superpoint: Self-supervised interest point detection and description. *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 224–236.

Engel, J., Koltun, V., Cremers, D., 2017. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3), 611–625.

Engel, J., Schöps, T., Cremers, D., 2014. Lsd-slam: Large-scale direct monocular slam. *European conference on computer vision*, Springer, 834–849.

Forster, C., Pizzoli, M., Scaramuzza, D., 2014. Svo: Fast semi-direct monocular visual odometry. *2014 IEEE international conference on robotics and automation (ICRA)*, IEEE, 15–22.

Godard, C., Mac Aodha, O., Brostow, G. J., 2017. Unsupervised monocular depth estimation with left-right consistency. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 270–279.

Grupp, M., 2017. evo: Python package for the evaluation of odometry and slam.

Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60, 91–110.

Mur-Artal, R., Tardós, J. D., 2017. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5), 1255–1262.

Rublee, E., Rabaud, V., Konolige, K., Bradski, G., 2011. Orb: An efficient alternative to sift or surf. *2011 International conference on computer vision*, Ieee, 2564–2571.

Sarlin, P.-E., DeTone, D., Malisiewicz, T., Rabinovich, A., 2020. Superglue: Learning feature matching with graph neural networks. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4938–4947.

Tateno, K., Tombari, F., Laina, I., Navab, N., 2017. Cnn-slam: Real-time dense monocular slam with learned depth prediction. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6243–6252.

Yang, C., Chen, Q., Yang, Y., Zhang, J., Wu, M., Mei, K., 2022. SDF-SLAM: a deep learning based highly accurate SLAM using monocular camera aiming at indoor map reconstruction with semantic and depth fusion. *IEEE Access*, 10, 10259–10272.

Yu, C., Liu, Z., Liu, X.-J., Xie, F., Yang, Y., Wei, Q., Fei, Q., 2018. Ds-slam: A semantic visual slam towards dynamic environments. *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE, 1168–1174.

Zhang, X., Dong, H., Zhang, H., Zhu, X., Li, S., Deng, B., 2025. A real-time, robust and versatile visual-SLAM framework based on deep learning networks. *IEEE Transactions on Instrumentation and Measurement*.

Zhao, Z., Wu, C., Kong, X., Li, Q., Guo, Z., Lv, Z., Du, X., 2025. Light-SLAM: A robust deep-learning visual SLAM system based on LightGlue under challenging lighting conditions. *IEEE Transactions on Intelligent Transportation Systems*.

Zou, Y., Luo, Z., Huang, J.-B., 2018. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. *Proceedings of the European conference on computer vision (ECCV)*, 36–53.

Zubizarreta, J., Aguinaga, I., Montiel, J. M. M., 2020. Direct sparse mapping. *IEEE Transactions on Robotics*, 36(4), 1363–1370.