Results in Engineering 30 (2026) 110864

Contents lists available at ScienceDirect

Results in Engineering

journal homepage: www.sciencedirect.com/journal/results-in-engineering

Research paper

Robust and low-complexity feature matching for UAV SLAM in repetitive environments

Cong Hoang Quach [a], Stuart Perry [a], Ha Vu Le [b], Thuy Pham [a], Thuan Hoang Tran [c], Manh Duong Phung [d,*]

[a] School of Electrical and Data Engineering, University of Technology Sydney, New South Wales, Australia
[b] University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam
[c] Duy Tan University, Da Nang, Vietnam
[d] College of Engineering and Computer Science and Smart Green Transformation Center (GREEN-X), VinUniversity, Hanoi, Vietnam

ARTICLE INFO

Keywords:
Image matching
Feature matching
Simultaneous localization and mapping
Unmanned aerial vehicle
Micro aerial vehicle
Inertial navigation

ABSTRACT

Perceptual aliasing remains a significant challenge for feature-based simultaneous localization and mapping (SLAM), especially in resource-constrained unmanned aerial vehicles (UAVs) operating in GPS-denied environments. This paper presents a low-complexity image-matching approach that improves correspondence reliability while maintaining computational efficiency. Feature matching is formulated as a linear assignment problem, where the cost matrix is designed to integrate descriptor similarity, feature distinctiveness estimation, and motion-consistent geometric constraints. Two practical pipelines are proposed: a VIO-guided strategy that incorporates visual-inertial priors, and an Instant Matching strategy that estimates pose directly from stereo observations when inertial information is unreliable. Experiments on 42 TartanAir sequences show that the proposed methods achieve comparable accuracy to state-of-the-art deep learning methods like SuperGlue while reducing computational cost by up to 10x and significantly lowering memory usage. These properties make the approach well suited for real-time deployment on micro-UAV platforms.

1. Introduction

Unmanned aerial vehicles (UAVs) have become increasingly important in a wide range of applications, including infrastructure inspection, environmental monitoring, search and rescue, precision agriculture, and delivery services [1–3]. Simultaneous localization and mapping (SLAM) is essential for the autonomous operation of unmanned aerial vehicles (UAVs), particularly in GPS-denied environments such as indoor spaces, urban canyons, and cluttered areas where satellite signals are unreliable. In such settings, the objective of SLAM is to enable a UAV to reconstruct a map of the surrounding environment while simultaneously estimating its own pose in real time using onboard sensors [4]. Due to strict constraints on payload, power consumption, and computational resources, lightweight CMOS image sensors are especially well suited for UAV navigation. Accordingly, the design of low-complexity, feature-based visual SLAM algorithms is crucial for enabling safe and reliable autonomous flight.

1.1. Feature-based visual SLAM

A visual SLAM system typically includes two key components: the front end and the back end. In feature-based visual SLAM, the front end detects and tracks the discrete 2D locations of feature points. Each 2D feature point needs to be associated with a specific 3D landmark in the environment in a process called data association [5]. This is a challenging task in SLAM due to the existence of noise, uncertainty, and ambiguity in the camera's measurements and the environment.

The back end reconstructs the 3D observation of feature points by using maximum a posteriori (MAP) estimation. For visual sensors, this process corresponds to bundle adjustment (BA) that minimizes the reprojection errors of feature points extracted by the front end. Such optimization enables consistent feature matching across multiple viewpoints over extended time windows. A prominent example is ORB-SLAM2 [6], which performs feature association using a covisibility graph and achieves accurate trajectory estimation across standard benchmarks.

* Corresponding author.
E-mail addresses: hoang.c.quach@student.uts.edu.au (C.H. Quach), Stuart.Perry@uts.edu.au (S. Perry), halv@vnu.edu.vn (H.V. Le), thuy.pham@uts.edu.au (T. Pham), tranthuanhoang@duytan.edu.vn (T.H. Tran), duong.pm@vinuni.edu.vn (M.D. Phung).

https://doi.org/10.1016/j.rineng.2026.110864
Received 25 February 2026; Received in revised form 19 April 2026; Accepted 3 May 2026
Available online 5 May 2026

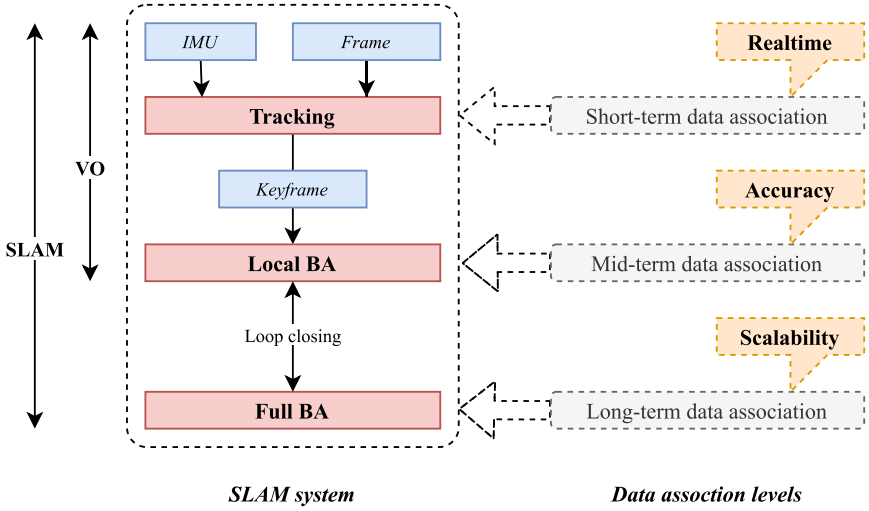


Fig. 1. A typical SLAM system with multiple types of data association: each stage has its priority, requiring a different type of data association. The short-term data association is the key for real-time estimation of SLAM; the mid-term determines the accuracy of trajectory and map model; the long-term provides loop closing ability and guarantees the consistency of the map model.

These results highlight that data association remains a major focus of SLAM research.

Data association in SLAM operates at multiple levels, as shown in Fig. 1. In advanced frameworks such as ORB-SLAM2, short-term data association can be extended to mid-term association to reduce drift accumulation. Although this strategy requires additional computational resources, mid-term data association improves consistency and can eliminate drift over extended trajectories, which is a key factor behind the high accuracy achieved by state-of-the-art SLAM systems.

1.2. Perceptual aliasing

Perceptual aliasing occurs when distinct locations in the environment produce similar sensory observations, leading to incorrect correspondences during data association [7]. This problem frequently arises when UAVs operate in environments with repetitive patterns, such as ceilings, floors, windows, trees, or large textureless regions like the sky. In practical applications, addressing perceptual aliasing is essential for maintaining reliable SLAM performance. A promising strategy is to integrate geometry-based constraints with learning-based methods to improve robustness. Although learning-based approaches have demonstrated strong performance in challenging scenarios [8], perceptual aliasing remains insufficiently addressed in most existing works [7,9]. Their high computational and memory demands also limit their applicability on resource-constrained UAVs. Consequently, identifying effective and efficient solutions to this problem remains an open research challenge.

In this work, we propose an image-matching approach to address perceptual aliasing in UAVs with limited computational resources. Feature matching is formulated as a linear assignment problem with a refined cost matrix to improve correspondence accuracy. Our main contributions are threefold:

(i) We introduce a feature distinctiveness estimation strategy to refine the matching cost matrix and incorporate geometric constraints to enhance robustness under challenging perceptual aliasing conditions.
(ii) We develop two matching pipelines for UAV applications, one leveraging visual-inertial odometry (VIO) priors and the other operating without VIO. Both achieve accuracy comparable to state-of-the-art methods [10].
(iii) The proposed algorithm maintains low computational complexity and memory consumption, enabling efficient deployment on resource-constrained platforms such as micro UAVs.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 describes the proposed methods. Experimental results are presented in Section 4. Finally, conclusions are provided in Section 5.

2. Related work

This section reviews representative data association methods for UAV-based SLAM that address perceptual aliasing. Considering the computational limitations of UAV platforms, we also discuss practical aspects, including low-complexity approaches and commonly used benchmark datasets.

2.1. Visual data association

Visual data association methods can be categorized into area-based and feature-based approaches [11]. Area-based methods are particularly effective for high-overlap imagery and close-range stereo settings [12–14]. Direct SLAM techniques adopt area-based matching for short-term data association and can perform well in textureless or motion-blurred scenarios. However, their memory and computational demands increase significantly as the number of associated frames grows, which limits their scalability to mid-term and long-term data association (see Fig. 1). Consequently, feature-based SLAM methods are often preferred in resource-constrained systems where computational efficiency is critical [15,16].

Feature-based matching involves feature detection, descriptor extraction, and correspondence estimation. Classical approaches employ conventional features such as SIFT, SURF, and ORB for detection and description. Correspondence estimation may rely on descriptor matching, graph-based optimization, or outlier rejection techniques, and can be extended to 3D associations. These methods are generally more flexible and efficient than area-based approaches and have been widely adopted in SLAM systems [17,18].

To improve robustness under challenging conditions [19], learning-based matching methods have been introduced. SuperGlue [10] is a representative example that estimates correspondences between two sets of interest points by solving a graph-based linear assignment problem.

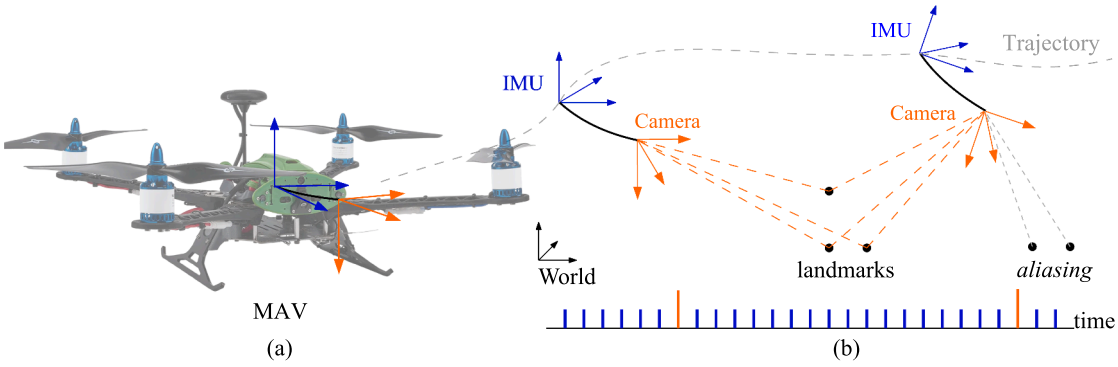


Fig. 2. Visual-inertial sensing on a MAV under perceptual aliasing. (a) A MAV equipped with a stereo visual-inertial sensor. Blue axes denote the IMU frame and orange axes the camera frame; the black curve in (b) illustrates their rigid transformation. (b) During flight, the IMU measures motion at a higher rate independent of scene appearance, while the camera observes scene landmarks. Black dots are landmarks; orange dashed lines indicate correct observations, and gray dashed lines represent mismatches due to perceptual aliasing.

Combined with SuperPoint descriptors [20], SuperGlue achieves state-of-the-art image matching performance on standard benchmarks and has become a widely adopted baseline in modern SLAM research. However, perceptual aliasing poses a significant challenge: visually similar regions with low image overlap can lead to ambiguous descriptors and incorrect correspondences [5], thereby degrading state estimation accuracy. Addressing perceptual aliasing through robust visual matching remains essential for reliable SLAM performance [21].

2.2. Low-complexity data association

Due to the real-time requirements of SLAM, low-complexity data association methods are widely adopted. For short-term association, the Lucas-Kanade tracker is commonly used as it operates on pixel intensities using area-based principles [22]. Feature-based matching approaches using SIFT or ORB descriptors combined with RANSAC have also become popular due to their robustness compared to purely intensity-based tracking [23]. However, RANSAC-based outlier rejection is sensitive to parameter selection and initialization, which may affect stability in challenging scenarios.

Deep learning-based matchers, such as SuperGlue, significantly improve correspondence accuracy. Nevertheless, in environments with large repetitive textures, attention mechanisms struggle to distinguish between similar patterns [10]. In such cases, the additional computational cost may outweigh the resulting accuracy gains.

An alternative practical solution is visual-inertial SLAM, which integrates measurements from an Inertial Measurement Unit (IMU) [24,25]. As illustrated in Fig. 2, IMUs operate at higher sampling rates than cameras and are independent of scene appearance. This makes them a valuable complement to visual sensing, helping to mitigate the effects of perceptual aliasing [26,27]. However, IMU-based motion estimation suffers from drift due to sensor biases [26]. Therefore, reliable feature-based visual matching remains essential to constrain drift and maintain long-term accuracy.

2.3. Datasets and practical challenges

Benchmark datasets play a critical role in SLAM research, as they provide standardized environments for evaluating accuracy, robustness, and computational efficiency. Constructing SLAM datasets requires careful consideration of sensor configuration, synchronization accuracy, operating environments, and targeted SLAM challenges. The EuRoC MAV dataset [28] is a widely adopted benchmark for VI SLAM on micro aerial vehicles (MAVs). It provides stereo images and IMU data with precise time synchronization and motion capture ground truth. The TUM VI dataset [29] focuses on challenges relevant to direct VI methods, including photometric calibration for stereo fisheye cameras. TartanAir [19] is a large-scale simulation dataset designed primarily for learning-based SLAM approaches. It offers diverse trajectories and visually complex scenes that enable evaluation under lighting changes, shadows, and perceptual aliasing. However, TartanAir does not provide IMU measurements, as it was not originally designed for a specific visual-inertial SLAM system or MAV application.

In practice, collecting real-world SLAM datasets poses additional difficulties. Commercial UAV platforms such as DJI and Skydio typically restrict low-level access to synchronized IMU and raw image streams, limiting reproducibility and system-level experimentation. Open research platforms that provide full sensor access exist [30–33], but they are often costly and require significant engineering effort to operate and maintain. Moreover, achieving centimeter-level ground truth, precise sensor calibration, and stable synchronization in outdoor environments remains a significant technical challenge.

For these reasons, photorealistic simulation tools such as AirSim [34] and Flightmare [35] have become practical alternatives for dataset construction. Simulation environments allow controlled evaluation of perceptual aliasing, motion difficulty, and sensor noise, while providing accurate ground-truth trajectories. Such tools are particularly valuable when studying data association robustness under challenging but repeatable conditions.

3. Methodology

In this work, we address perceptual aliasing challenges in SLAM systems for UAVs using low-complexity image-matching algorithms. Modern UAV platforms typically integrate multiple sensing modalities, including GPS, IMU, and stereo cameras, to perform visual-inertial odometry (VIO). Building on this architecture, we propose two complementary matching strategies for practical deployment:

- **Method 1: VIO Matching**, which utilizes the UAV's visual-inertial odometry estimate as prior information to guide feature correspondence.
- **Method 2: Instant Matching**, which operates independently of the VIO output and estimates pose directly from stereo visual information.

As illustrated in Fig. 3, both methods follow a feature-based pipeline using SuperPoint for keypoint detection and descriptor extraction. The detected features are used to construct a cost map, which is refined through our proposed visual and geometric constraints before final assignment. The resulting correspondences are then passed to the MAP estimation module to update the global state of the SLAM system.

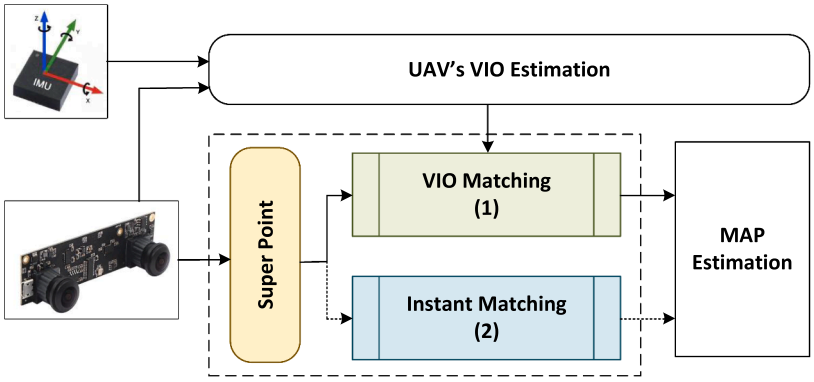


Fig. 3. Proposed matching framework within a UAV SLAM system. The green block (1) represents *VIO Matching*, and the blue block (2) represents *Instant Matching*. Both operate within the data association module using SuperPoint [20] features. VIO Matching utilizes visual-inertial priors to guide correspondence selection, while Instant Matching relies solely on visual cues when VIO estimates are unreliable. The selected matches are then used for maximum a posteriori (MAP) state estimation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

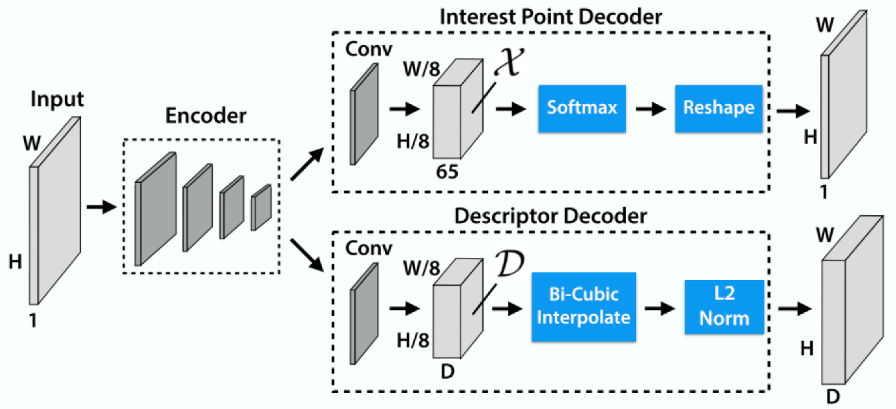


Fig. 4. SuperPoint network architecture [20]. A fully convolutional network with a shared encoder and two decoder heads for keypoint detection and descriptor extraction. The encoder reduces spatial resolution, while the decoders generate probability maps and normalized descriptors. Gray blocks denote learnable layers, and blue blocks represent non-learnable upsampling and normalization operations. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

While Method 1 generally achieves higher accuracy with lower computational cost when reliable VIO information is available, Method 2 provides a robust alternative under severe perceptual aliasing, where the VIO estimate becomes degraded. The complete processing pipelines of the two methods are illustrated in Figs. 6 and 7, and are described in detail below.

3.1. SuperPoint feature detector and descriptor

SuperPoint is a fully convolutional neural network that jointly performs 2D keypoint detection and descriptor extraction in a single forward pass. As illustrated in Fig. 4, the architecture consists of a shared VGG-style encoder [36] followed by two parallel decoder heads: an interest point decoder and a descriptor decoder.

Unlike traditional feature-based detectors such as ORB, SuperPoint does not explicitly enforce semantic structures such as corners or line endpoints. Instead, it learns interest point locations directly from data, which improves robustness to illumination variation, lens distortion, and viewpoint changes. This property is particularly advantageous for UAV applications using wide field-of-view or fisheye cameras, as SuperPoint can operate directly on raw images without additional preprocessing steps such as histogram equalization. As a result, it supports low-latency feature extraction and contributes to efficient real-time SLAM state estimation.

3.2. Feature matching

The keypoints and descriptors extracted by SuperPoint are used for inter-frame data association in the SLAM pipeline. This matching process can be formulated as a cost minimization problem based on descriptor similarity.

Let $\{\mathbf{d}_i^s\}_{i=1}^m$ and $\{\mathbf{d}_j^t\}_{j=1}^n$ denote the feature descriptors extracted from the source and target images, respectively. The cross L2-norm cost matrix $C \in \mathbb{R}^{m\times n}$ is defined as

$$C_{ij} = \|\mathbf{d}_i^s - \mathbf{d}_j^t\|_2, \tag{1}$$

where C_{ij} represents the dissimilarity between the ith source feature and the jth target feature. A smaller value of C_{ij} indicates higher descriptor similarity and therefore a stronger candidate correspondence.

In conventional one-to-one correspondences, a binary assignment matrix $X \in \{0,1\}^{m\times n}$ is defined such that

$$X_{ij} = \begin{cases} 1, & \text{if source feature } i \text{ is matched to target feature } j, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The feature matching problem can then be formulated as the following linear assignment problem:

$$\min_X \sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij}X_{ij} \tag{3}$$

subject to the one-to-one matching constraints

$$\sum_{j=1}^{n} X_{ij} \le 1, \quad \forall i = 1, \dots, m, \tag{4}$$

$$\sum_{i=1}^{m} X_{ij} \le 1, \quad \forall j = 1, \dots, n, \tag{5}$$

$$X_{ij} \in \{0, 1\}. \tag{6}$$

The first two constraints ensure that each source feature is matched to at most one target feature and vice versa. However, if the environment contains perceptual aliasing, extracted feature points have similar descriptors and are hard to distinguish by directly solving assignment Eq. (3). In this situation, there are many point pairs with small L2 norm distances, which can be visualized as black dots in the cost-map matrix as shown in Fig. 5b. This leads to wrong alignments as shown in the red lines in Fig. 5c. To address this issue, we introduce a cost matrix regularization framework that incorporates feature distinctiveness and global consistency before final discrete matching.

3.3. Feature distinctiveness estimation

Conventional matching assumes all features are equally reliable. However, in repetitive environments, many features are ambiguous and should not contribute equally to the matching process. We therefore estimate feature distinctiveness within each image using intra-image descriptor similarity.

Let $D = \{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_N\}$ denote the descriptors extracted from a single image. We compute the self-similarity matrix

$$L2_{self}(i, j) = \|\mathbf{d}_i - \mathbf{d}_j\|_2, \tag{7}$$

which measures pairwise descriptor distances within the same image. For each feature, the average distance to all other features reflects its uniqueness: a larger mean indicates a distinctive feature, while a smaller mean indicates ambiguity due to perceptual aliasing.

Based on this observation, we define uniqueness scores:

$$H_{\text{source}} = \max(0, \text{mean}(L2_{self\_source}) - 1), \tag{8a}$$

$$H_{\text{target}} = \max(0, \text{mean}(L2_{self\_target}) - 1), \tag{8b}$$

where subtracting 1 acts as a threshold to suppress low-uniqueness features. Because SuperPoint descriptors are L2-normalized, a Euclidean distance of 1 corresponds to a cosine similarity of 0.5. We therefore use this value as a baseline threshold to suppress highly clustered and weakly distinctive features. The max operation ensures non-negative values.

3.4. Soft matching using uniqueness scores

Incorporating the uniqueness scores, we relax the discrete assignment constraint in (3) and reformulate the matching problem as a soft optimal transport model. Specifically, we introduce a soft assignment matrix $P \in \mathbb{R}_{+}^{m \times n}$, where $P_{ij} \ge 0$ represents the transport mass between source feature i and target feature j. The relaxed matching problem becomes

$$\min_{P \ge 0} \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} P_{ij}, \tag{9}$$

where C_{ij} is the cross L2-norm cost. To improve numerical stability and avoid unstable solutions, we introduce entropic regularization:

$$\min_{P \ge 0} \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} P_{ij} + \lambda \sum_{i=1}^{m} \sum_{j=1}^{n} P_{ij} \log P_{ij}, \tag{10}$$

where $\lambda > 0$ controls the smoothness of the solution.

The entropic optimal transport problem in (10) can be efficiently solved using the Sinkhorn-Knopp algorithm. The algorithm first converts the cost matrix into a similarity kernel

$$K = \exp(-C/\lambda), \tag{11}$$

and then alternates between row and column normalizations.

In the standard Sinkhorn formulation, rows and columns are normalized to sum to one:

$$K_{ij} \leftarrow K_{ij} \cdot \frac{1}{\sum_j K_{ij}}, \qquad K_{ij} \leftarrow K_{ij} \cdot \frac{1}{\sum_i K_{ij}}. \tag{12}$$

However, this assumes all features are equally reliable. We instead normalize using the uniqueness scores:

$$K_{ij} \leftarrow K_{ij} \cdot \frac{H_{\text{source}}(i)}{\sum_j K_{ij}}, \qquad K_{ij} \leftarrow K_{ij} \cdot \frac{H_{\text{target}}(j)}{\sum_i K_{ij}}. \tag{13}$$

This modification allows distinctive features to carry greater matching mass while suppressing ambiguous ones. After convergence, the resulting transport matrix is denoted as

$$G_s = [K_{ij}]. \tag{14}$$

It represents a globally consistent, distinctiveness-weighted soft correspondence matrix. Compared to the naive cost matrix in (1), G_s provides a more structured representation for subsequent geometric refinement and discrete matching.

3.5. Geometric constraints

Descriptor similarity alone is insufficient for reliable feature matching in environments with strong perceptual aliasing. To improve robustness, we incorporate geometric constraints that enforce physical consistency with the estimated motion of the UAV. In the scenario where visual-inertial odometry (VIO) is available, the relative pose between consecutive camera frames can be estimated and exploited to guide the matching process. Given this estimated transformation between the source and target camera frames, each source keypoint $\mathbf{p}_i^s$ detected in the source image is projected into the target image plane using the camera intrinsics and the estimated pose. This projection yields an expected 2D location $\hat{\mathbf{p}}_i^t$ in the target image that represents where the source keypoint should appear if the correspondence is geometrically correct.

Let $\mathbf{p}_j^t$ denote a candidate keypoint detected in the target image. The geometric consistency between $\mathbf{p}_i^s$ and $\mathbf{p}_j^t$ is evaluated by measuring the Euclidean distance between the predicted location $\hat{\mathbf{p}}_i^t$ and the observed keypoint $\mathbf{p}_j^t$ in the target image. This reprojection-based distance provides a direct measure of how well the candidate correspondence agrees with the estimated camera motion. The resulting geometric cost is defined as:

$$D_{ij} = \sqrt{\frac{\left\|\hat{\mathbf{p}}_i^t - \mathbf{p}_j^t\right\|_2}{d}}, \tag{15}$$

where d is a normalized parameter chosen according to the resolution of the input images. Specifically, d is defined as the diagonal length of the image, $d = \sqrt{W^2 + H^2}$, where W and H represent the image width and height in pixels, respectively. The square-root operation moderates the influence of large reprojection errors and prevents the geometric term from dominating the overall matching cost.

The geometric distance D_{ij} serves as an additional constraint that complements appearance-based similarity. When integrated with the Sinkhorn-regularized matching matrix, this term suppresses false matches that are inconsistent with the camera geometry. The final cost matrix, C^*, considering both appearance and motion cues is defined as:

$$C^* = (1 - G_s) + D. \tag{16}$$

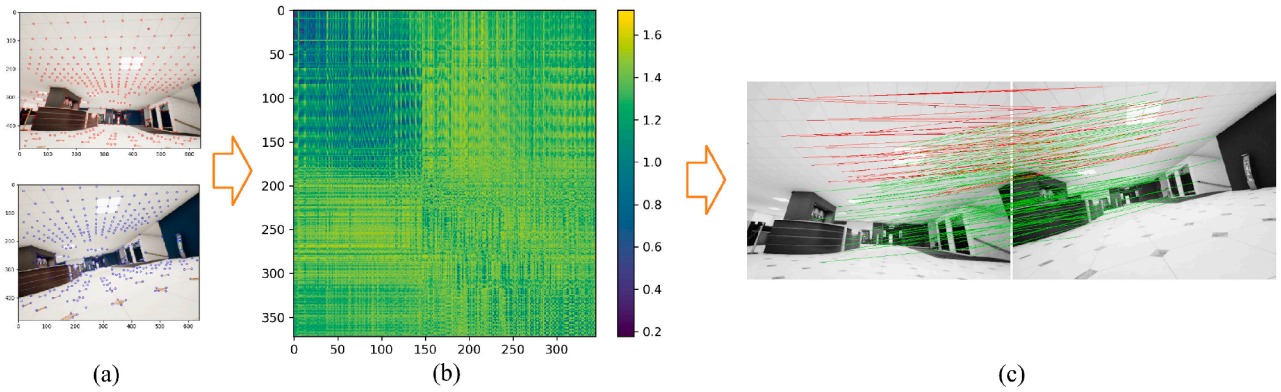


Fig. 5. Effect of perceptual aliasing on naive Hungarian matching. (a) Source and target images with detected feature points. (b) L2-norm cost matrix computed between all feature pairs; darker (blue) values indicate smaller descriptor distances. Repetitive ceiling patterns produce a large low-cost region in the upper-right area of the matrix, reflecting perceptual aliasing. (c) Resulting correspondences using naive Hungarian matching: green lines denote correct matches, while red lines indicate mismatches caused by aliasing. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

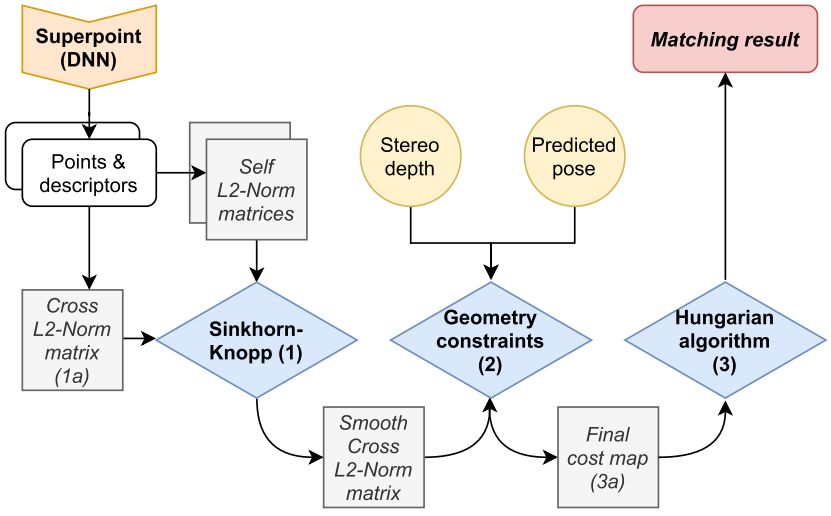


Fig. 6. Method 1 - VIO Matching. Rather than directly optimizing the raw L2-norm cost matrix (1a), the method first applies Sinkhorn-based regularization (1) and then incorporates geometric constraints (2) to obtain the refined cost matrix (3a). Circular blocks represent prior information from stereo depth and VIO pose estimation; blue diamond blocks denote the main computational modules; and square blocks indicate intermediate data produced during processing. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The Hungarian method optimizes the cost matrix C^* and gives the final matching result.

3.6. Instant matching without VIO

In practice, the pose predicted by VIO may be inaccurate or completely unavailable, particularly when the UAV undergoes rapid motion or operates in environments with strong perceptual aliasing. In such cases, relying on VIO-based reprojection can introduce significant geometric errors and degrade matching performance. To handle this situation, we propose an alternative strategy that estimates the relative pose using stereo images alone.

As illustrated in Fig. 7, the key difference of this method lies in the additional pose-estimation steps inserted before enforcing geometric constraints. After obtaining the Sinkhorn-regularized cost matrix G_s, an initial set of tentative correspondences is extracted using the Hungarian algorithm:

$$M_{\text{raw}} = \text{Hungarian}(1 - G_s), \tag{17}$$

where M_{raw} denotes the raw matching result prior to geometric verification.

Using stereo depth information, each matched feature provides a 3D-2D correspondence pair $(\mathbf{X}_i, \mathbf{p}_i^t)$, where $\mathbf{X}_i \in \mathbb{R}^3$ is the reconstructed 3D point in the source frame and $\mathbf{p}_i^t$ is its 2D observation in the target image. The relative camera pose $(\mathbf{R}, \mathbf{t})$ is then estimated by solving a Perspective-n-Point (PnP) problem under RANSAC:

$$(\mathbf{R}, \mathbf{t}) = \arg\min_{\mathbf{R},\mathbf{t}} \sum_{i \in \mathcal{I}} \left\| \mathbf{p}_i^t - \pi(\mathbf{R}\mathbf{X}_i + \mathbf{t}) \right\|_2^2, \tag{18}$$

where $\pi(\cdot)$ denotes the camera projection function and $\mathcal{I}$ is the inlier set determined by RANSAC. This optimization yields a coarse 3D pose estimate derived purely from stereo observations, which serves as an anchor replacing the missing or unreliable VIO estimate.

Once a valid pose $(\mathbf{R}, \mathbf{t})$ is recovered, we perform pose-guided reprojection to compute the geometric consistency term:

$$D_{ij} = \sqrt{\frac{\left\| \hat{\mathbf{p}}_i^t - \mathbf{p}_j^t \right\|_2}{d}}, \qquad \hat{\mathbf{p}}_i^t = \pi(\mathbf{R}\mathbf{X}_i + \mathbf{t}). \tag{19}$$

This term is then integrated with the appearance-based cost matrix to form the refined cost map:

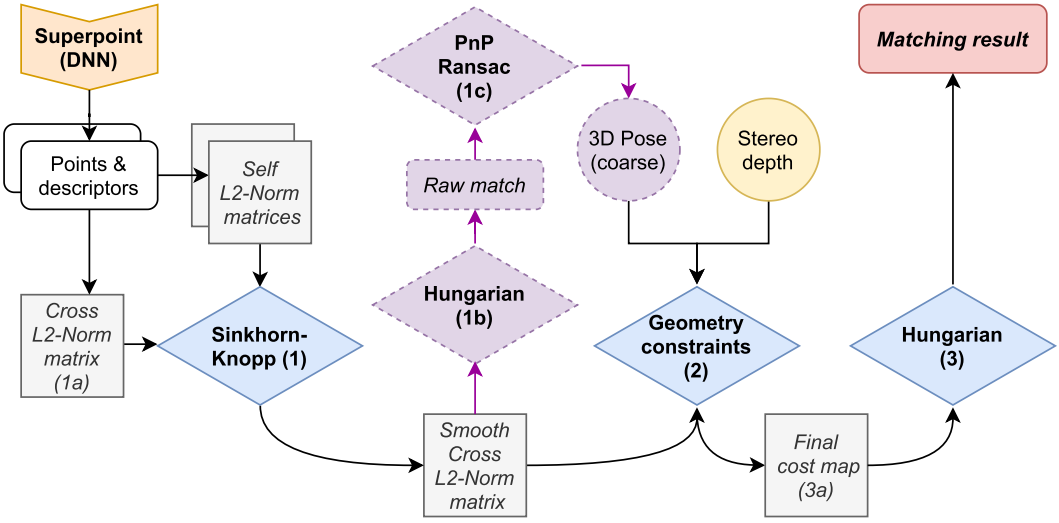


Fig. 7. Method 2 - Instant Matching. Sinkhorn regularization (1) and geometric constraints (2) are used to obtain the refined cost matrix (3a). When VIO pose is unavailable, initial matches are extracted by Hungarian (1b) and a coarse pose is estimated via RANSAC-PnP (1c) (purple blocks) before final matching. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$C^* = (1 - G_s) + D. \tag{20}$$

Finally, the geometry-aware cost map C^* is optimized using the Hungarian algorithm to produce the final matching result. By explicitly estimating the pose from stereo observations, this method maintains robust correspondence estimation even when inertial information is unreliable.

3.7. System-level switching logic

To integrate the proposed matching pipelines into an autonomous UAV software stack, the SLAM system requires a deterministic mechanism to evaluate VIO reliability in real time and trigger transitions between Method 1 and Method 2.

Method 1 (VIO Matching) serves as the default operational state as it provides the most computationally efficient and geometrically constrained correspondences when the vehicle dynamics are stable. To continuously monitor the reliability of the VIO prior, we evaluate two primary system metrics at each frame:

1. Tracking inlier count (N_{trk}): The number of valid feature correspondences successfully tracked from the previous frame. A severe drop in N_{trk} indicates highly dynamic motion, motion blur, or sudden view occlusion.
2. State uncertainty (Σ_{IMU}): The trace of the IMU preintegration covariance matrix. A rapid growth in covariance indicates that the IMU biases are diverging or that the inertial measurements are poorly constrained by recent visual updates.

The VIO estimate is considered unreliable if N_{trk} falls below a minimum safe threshold or if the state uncertainty exceeds a predefined stability bound. When either trigger condition is met, the system automatically transitions to Method 2 (Instant Matching).

4. Results

We have conducted a number of experiments to evaluate the performance of the proposed image-matching methods as detailed below.

4.1. Experimental setup

4.1.1. Synthetic data generation

The proposed methods were evaluated on the *Offices* and *CarWelding* sequences from the TartanAir dataset, which contain perceptual aliasing due to repetitive visual structures. Since TartanAir does not provide IMU measurements required for VIO, we performed additional steps to generate synthetic IMU data compatible with our SLAM framework.

First, timestamps were assigned to the ground-truth camera poses to construct a temporally consistent trajectory. Using OpenVINS [37], the ground-truth trajectory was interpolated via B-spline fitting. Synthetic IMU measurements, including gyroscope and accelerometer readings, were subsequently generated.

To evaluate matching performance under realistic UAV motion and perceptual aliasing conditions, we conducted experiments using image pairs separated by one second. Intermediate frames between the selected images were used to reconstruct the VIO trajectory through multi-state constraint Kalman filter (MSCKF) in OpenVINS.

4.1.2. Ground truth generation and automatic evaluation

Ground-truth correspondences are generated automatically by projecting 3D keypoints from the source image into the camera plane of the target image using the known ground-truth pose and depth information. Each ground-truth match is represented in 3D space as (i, j, d), where i and j denote the row and column pixel coordinates, respectively, and d represents the projected depth in the target view.

A predicted correspondence is considered correct if it satisfies both spatial and depth consistency criteria. Specifically, the reprojection error must be less than 8 pixels, and the depth error must be within 10% of the corresponding ground-truth depth value. The 8-pixel threshold is chosen to align with the 8×8 sampling window used in SuperPoint, while the 10% depth tolerance reflects the typical stereo depth accuracy within approximately $40\times$ the stereo baseline.

To evaluate matching performance, we adopt the F1 score as the primary metric for each image pair. Since SLAM performance is evaluated over entire image sequences, we compute the F1 score across all image pairs within a sequence. The final sequence-level F1 score is defined as the average F1 score of all evaluated image pairs.

4.1.3. Implementation details

Table 1 summarizes the key algorithmic and environmental parameters used in our implementation. The matching threshold controls the final discrete assignment stage, while the Sinkhorn parameters (λ and the number of iterations) determine the trade-off between entropic regularization and computational efficiency. For Instant Matching (Method 2), depth screening limits the 3D points to a reliable triangulation range

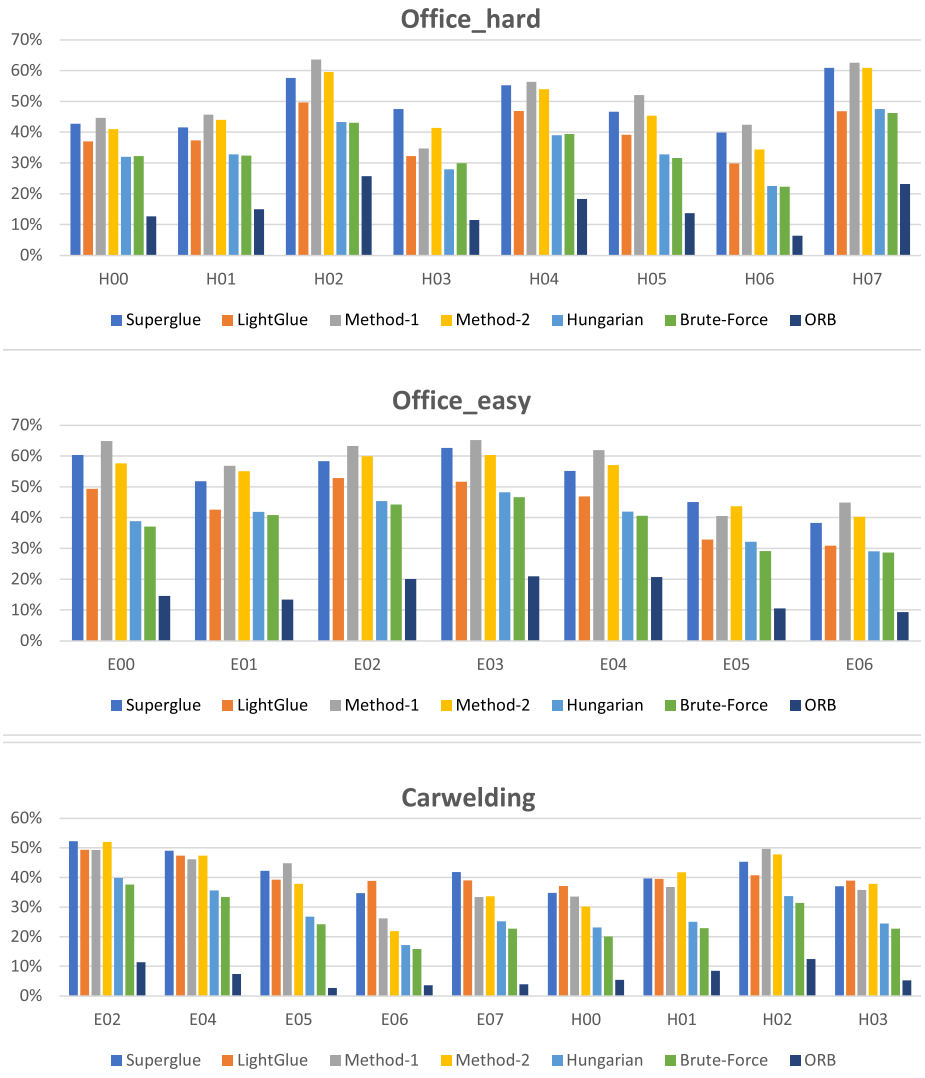


Fig. 8. F1-score comparison on TartanAir sequences. Matching accuracy (F1 score) across Office_easy, Office_hard, and CarWelding scenarios. Higher values indicate better matching performance.

Table 1
Algorithmic parameters used across all experiments.

Category	Parameter	Value
Feature Extraction	Number of keypoints (m, n)	250
	Descriptor dimension	256
Uniqueness Scoring	Distance metric	$L2$ norm
	Uniqueness threshold	1.0
Sinkhorn Algorithm	Regularization coefficient (λ)	0.05
	Iteration count	20
	Matching threshold	0.2
Geometric Constraints	Normalization distance (d)	$\sqrt{W^2 + H^2}$
	Effective depth screening	0.5 m – 20.0 m
	Maximum reprojection error	8.0 pixels
PnP-RANSAC (Method 2)	Inlier threshold	8.0 pixels
	Confidence level	0.99
	Maximum iterations	1000

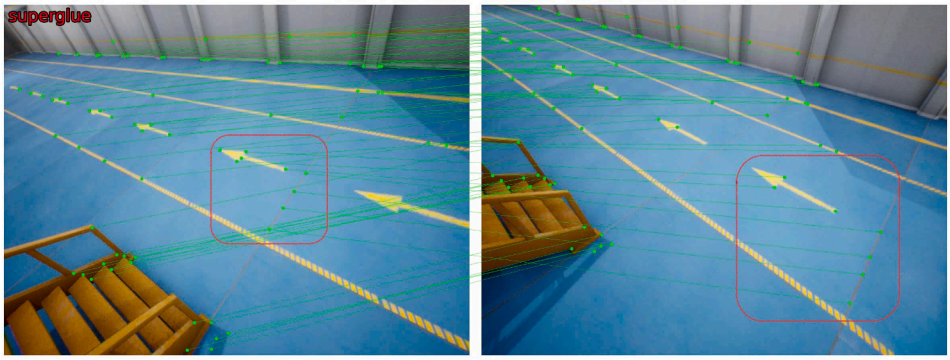
determined by the stereo baseline. The RANSAC thresholds are selected to tolerate minor calibration noise while rejecting geometric outliers.

Table 2
Matching accuracy (F1 score) across different scenarios on the TartanAir dataset.

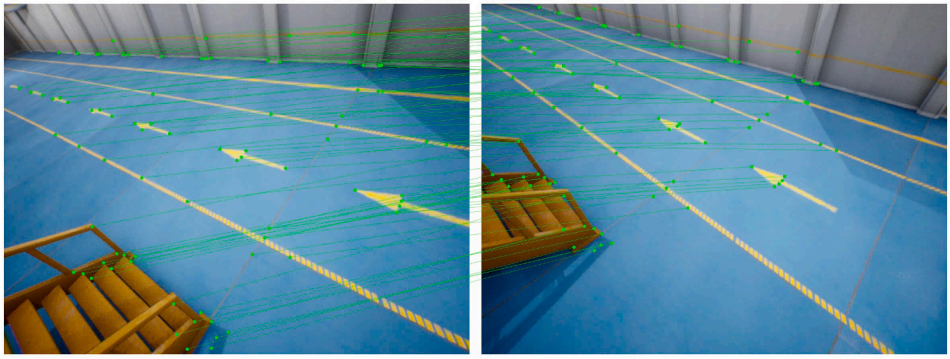
Method	office_ Easy	office_ Hard	office2_ Easy	office2_ Hard	carwelding
SuperGlue	53.07 ± 8.77	48.99 ± 7.93	58.29 ± 10.89	46.27 ± 8.41	41.85 ± 8.13
LightGlue	43.84 ± 8.86	39.84 ± 7.23	49.01 ± 10.90	39.53 ± 6.63	**42.83 ± 6.74**
Hungarian	39.63 ± 6.89	34.73 ± 8.14	44.41 ± 12.89	34.7 ± 9.31	27.87 ± 8.88
Brute-force	38.15 ± 7.00	34.64 ± 7.76	42.32 ± 13.29	32.84 ± 7.42	25.65 ± 8.4
ORB + MNN	15.64 ± 4.92	15.78 ± 6.34	16.97 ± 9.23	12.92 ± 5.8	7.67 ± 4.47
Method-1	**56.76 ± 10.07**	**50.24 ± 10.18**	**59.70 ± 11.42**	**47.4 ± 7.76**	39.48 ± 8.25
Method-2	53.41 ± 8.06	47.56 ± 9.51	54.95 ± 12.21	43.09 ± 11.2	38.9 ± 10.11

4.2. Matching performance evaluation

Table 2 and Fig. 8 summarize the matching performance across 42 sequences from the TartanAir dataset. The evaluation compares the proposed Method 1 (VIO Matching) and Method 2 (Instant Matching) with two learning-based matchers, SuperGlue [10] and LightGlue [38], as well as three classical low-complexity baselines, Hungarian, Brute-force,



(a) Feature matching by SuperGlue



(b) Feature matching by our VIO matching method

Fig. 9. Feature matching results in a repetitive environment. (a) SuperGlue produces incorrect correspondences when relying solely on descriptor similarity (highlighted in red). (b) The proposed VIO-based method achieves more geometrically consistent matches by incorporating VIO information. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

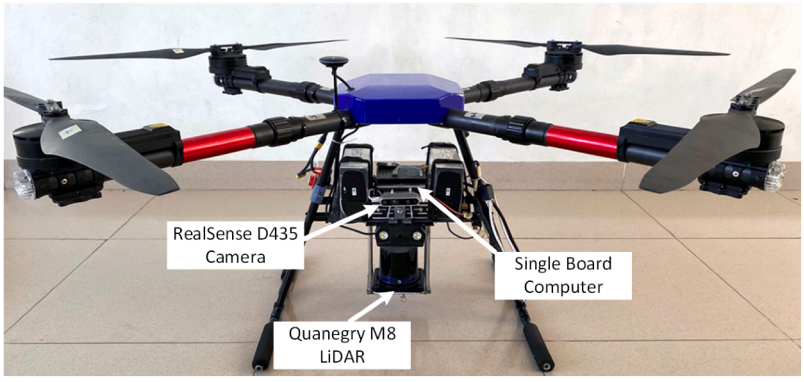
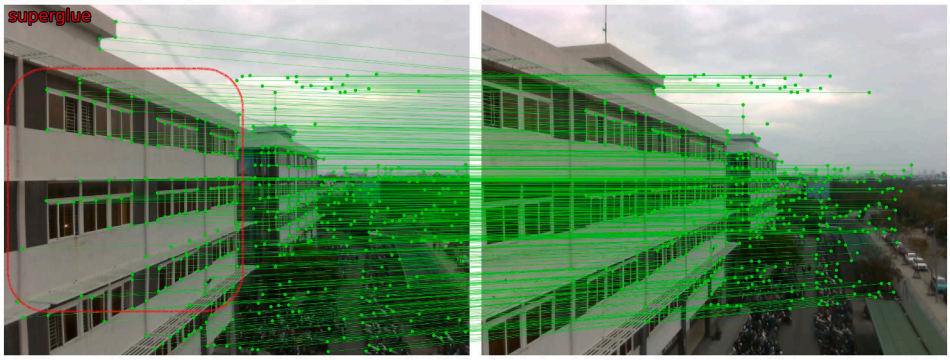


Fig. 10. UAV platform used for real-world data collection. The platform is equipped with a RealSense D435 camera, a Quanergy M8 LiDAR, and a single-board computer.

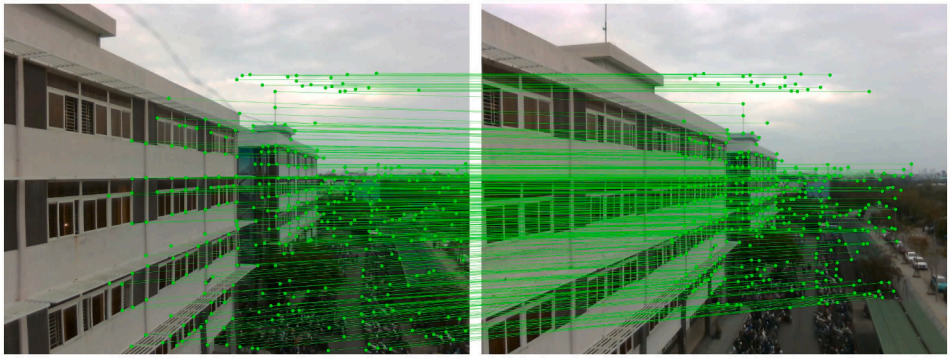
and ORB with Mutual Nearest Neighbor (ORB + MNN) matching [39], which are commonly used in SLAM systems [40,41].

Office Scenarios. In the *office_Easy* and *office2_Easy* sequences, Method 1 consistently achieves the highest F1 scores among the lightweight approaches, reaching 56.76% and 59.70%, respectively. These results outperform both SuperGlue (53.07% and 58.29%) and LightGlue (43.84% and 49.01%), and show a substantially larger improvement over the classical Hungarian, Brute-force, and ORB + MNN baselines. Method 2 also produces competitive results, closely matching SuperGlue in easy indoor environments while maintaining lower computational complexity, whereas ORB + MNN performs poorly in these repetitive scenes with F1 scores below 17%.

In the more challenging *office_Hard* and *office2_Hard* sequences, which exhibit stronger perceptual aliasing and larger viewpoint variations, Method 1 maintains stable performance (50.24% and 47.40%), remaining better than both SuperGlue (48.99% and 46.27%) and LightGlue (39.84% and 39.53%). Method 2 also remains competitive with scores of 47.56% and 43.09%, while Hungarian, Brute-force, and ORB + MNN degrade much more significantly. These results confirm that incorporating feature uniqueness and geometric constraints improves robustness in repetitive indoor environments.



(a) Matching results using SuperGlue.



(b) Matching results using the proposed VIO-based method.

Fig. 11. Feature matching on real-world data. (a) SuperGlue produces mismatches in repetitive regions (red box). (b) The proposed VIO-based method yields more geometrically consistent correspondences. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

CarWelding Scenario. The *CarWelding* sequences present challenging conditions due to dynamic objects, specular surfaces, and rapid illumination changes. All methods experience performance degradation in this scenario. LightGlue achieves the highest average F1 score (42.83%), followed closely by SuperGlue (41.85%), while Method 1 and Method 2 obtain 39.48% and 38.90%, respectively. Although the two learning-based methods provide slightly better accuracy in this case, the performance gap remains relatively small compared with their substantially higher computational cost. In contrast, Hungarian and Brute-force matching exhibit significantly lower performance (27.87% and 25.65%), and ORB + MNN performs worst overall at only 7.67%, which highlights the limitations of purely descriptor-based or naive assignment approaches under dynamic and visually ambiguous conditions.

Overall Observations. Across all scenarios, the proposed methods consistently outperform traditional low-complexity matching techniques and achieve performance comparable to SuperGlue, particularly in structured indoor environments. Compared with LightGlue, the proposed methods also provide noticeably stronger performance in the office scenarios, although LightGlue attains the best result in the CarWelding sequence. Method 1 demonstrates the strongest overall robustness due to its integration of VIO-based geometric priors, while Method 2 provides a reliable alternative when inertial information is unavailable or unreliable.

The results further suggest that point-level descriptor similarity alone is insufficient for handling highly dynamic and repetitive scenes, as shown in Fig. 9. This is particularly evident from the weak performance of ORB + MNN across all scenarios and from the reduced accuracy of Hungarian and Brute-force matching in the more challenging sequences. Even advanced learning-based matchers show limited gains under these conditions, indicating that additional scene-level reasoning or temporal consistency mechanisms may be required to further improve robustness.

Overall, the proposed approaches achieve a favorable balance between accuracy and computational efficiency, which makes them well suited for real-time UAV-based SLAM applications.

4.3. Real-world data validation

To validate the proposed framework in real-world urban environments, data were collected using a quadcopter platform equipped with a RealSense D435 camera, a Quanergy M8 LiDAR, and a single-board computer, as shown in Fig. 10. The feature matching results are presented in Fig. 11. Both the proposed method and SuperGlue detect and match a substantial number of feature points. However, SuperGlue produces incorrect correspondences in repetitive structures (e.g., building windows) due to the lack of motion and geometric priors. In contrast, the proposed method leverages VIO information to enforce geometric consistency and result in more reliable matching performance.

4.4. Computational complexity

To analyze computational efficiency, we estimate the number of floating-point operations (FLOPs) required by each method [42]. In practical visual SLAM systems, approximately 200–300 keypoints per image are typically sufficient for robust pose estimation, while using more than 1000 keypoints rarely provides additional benefit [22]. In our analysis, we assume 250 keypoints per image ($m = n = 250$), resulting in 500 total feature points per image pair, and 20 Sinkhorn iterations. Since SuperPoint feature extraction is performed in all compared methods, its

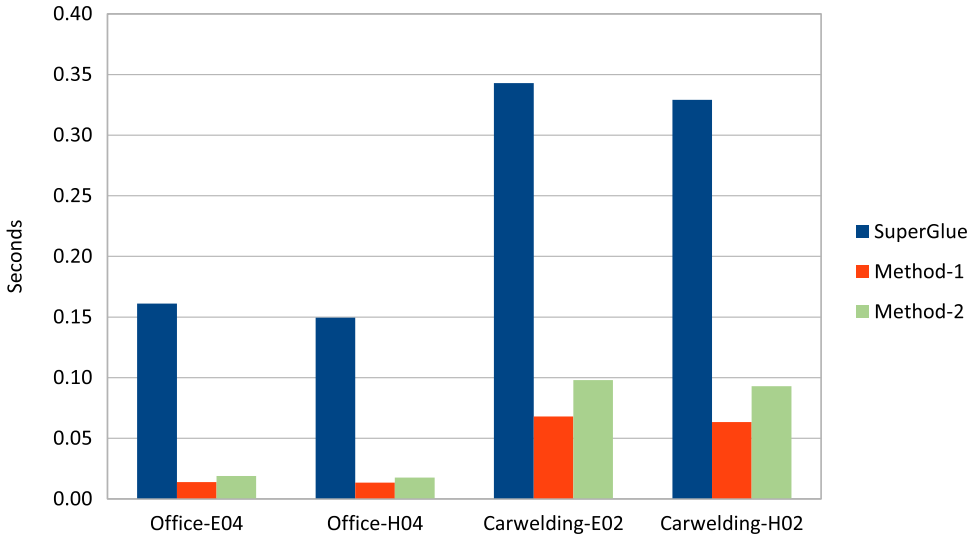


Fig. 12. CPU runtime comparison. Average computation time per image pair (seconds); lower values indicate better efficiency.

Table 3
Computational complexity in FLOPs. The estimates assume 250 feature points per image ($m = n = 250$) and 20 Sinkhorn iterations.

	Proposal methods		Superglue
	Sinkhorn-Knopp	Hungarian	
FLOPs	20 x 62.375 MFLOPS	62.5 MFLOPS	
	1.31 GFLOPs		22.1 GFLOPs

computational cost is treated as a shared front-end component and is therefore excluded from the FLOPs calculation.

SuperGlue and LightGlue complexity. SuperGlue employs a deep neural network with approximately 12 million parameters. A single forward pass requires approximately 22.1 GFLOPs, as summarized in Table 3. This computational demand arises primarily from attention layers and feature aggregation operations. LightGlue follows a similar matching paradigm but adopts a more efficient adaptive design. According to [38], the full LightGlue model is approximately 35% faster than SuperGlue.

Proposed method complexity. The computational cost of the proposed method is dominated by the Hungarian algorithm and the Sinkhorn-Knopp iterations. The Hungarian algorithm has cubic complexity, $\mathcal{O}(n^3)$. For $n = 250$, its total cost across all steps is approximately 62.5 MFLOPs. Each Sinkhorn iteration requires matrix scaling and multiplication operations with complexity proportional to $2n^2(2n - 1)$. For $n = 250$, this corresponds to approximately 62.4 MFLOPs per iteration. With 20 iterations, the total Sinkhorn cost is approximately

$$20 \times 62.4 \text{ MFLOPs} \approx 1.25 \text{ GFLOPs}.$$

Combining Hungarian and Sinkhorn operations yields an overall computational cost of approximately 1.31 GFLOPs, which is nearly 17 times lower than that of SuperGlue. For Method 2, the additional PnP-RANSAC module contributes a negligible overhead of less than 0.05 GFLOPs and therefore does not affect the overall complexity comparison.

Runtime evaluation. Fig. 12 reports the average CPU execution time measured on an x86 platform equipped with an Intel Core i7-12700H processor and 16 GB of RAM. To ensure a fair and deterministic comparison, the deep learning models were executed using PyTorch, and all algorithms were configured for single-threaded CPU execution. Under these controlled conditions, the proposed methods operate approximately 5–10 times faster than SuperGlue. The runtime difference varies with the number of detected keypoints. For example, the Office sequences contain roughly 500 total feature points per pair, while the CarWelding sequences may contain around 1000 feature points, leading to increased computation. Notably, more than 90% of the runtime in the proposed method is spent on matrix multiplications within the Sinkhorn iterations. Therefore, GPU implementation could further accelerate performance due to the high parallelizability of these operations.

Memory footprint. The proposed method requires approximately 100 MB of memory, significantly lower than SuperGlue's memory consumption of roughly 1 GB. The proposed low-complexity formulation is therefore better suited for real-time deployment on resource-constrained UAV platforms.

5. Conclusion

We presented a low-complexity feature matching framework that addresses perceptual aliasing in UAV SLAM. By integrating uniqueness scores with geometric constraints, our method effectively suppresses ambiguous features caused by repetitive textures. We proposed two pipelines, VIO Matching and Instant Matching, to ensure robust operation whether onboard odometry is reliable or unavailable.

Experiments on the TartanAir dataset demonstrate that our approach achieves matching accuracy comparable to state-of-the-art deep learning methods like SuperGlue while being approximately 10x faster. This efficiency makes the proposed framework highly suitable for resource-constrained UAVs operating in GPS-denied environments.

However, the current boundaries of the method's applicability should also be acknowledged. As shown by the results on the *CarWelding* sequences, although the proposed approach performs well in structured repetitive environments such as indoor offices and urban corridors, its performance becomes more limited in the presence of strong dynamics, severe specular reflections, and large appearance variations. In such highly unstructured or rapidly changing scenarios, the reliance on appearance-based distinctiveness and simple geometric priors is less robust than more heavily parameterized deep learning models. Future work will therefore focus on defining fallback mechanisms for these extreme conditions, applying the uniqueness metric to loop closure detection, and benchmarking the system on a broader range of datasets.

CRediT authorship contribution statement

Cong Hoang Quach: Writing – original draft, Visualization, Software, Methodology, Formal analysis, Data curation, Conceptualization; **Stuart Perry:** Writing – review & editing, Validation, Supervision, Resources, Methodology, Investigation, Conceptualization; **Ha Vu Le:** Writing – review & editing, Validation, Supervision, Resources, Methodology, Investigation, Conceptualization; **Thuy Pham:** Writing – review & editing, Resources, Project administration; **Thuan Hoang Tran:** Resources, Data curation; **Manh Duong Phung:** Writing – review & editing, Validation, Supervision, Resources, Methodology, Formal analysis, Data curation, Conceptualization.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Y. Yan, F. Song, J. Sun, The application of UAV technology in maize crop protection strategies: a review, Comput. Electron. Agric. 237 (2025) 110679.

[2] M.D. Phung, Q.P. Ha, Motion-encoded particle swarm optimization for moving target search using UAVs, Appl. Soft Comput. 97 (2020) 106705. https://doi.org/10.1016/j.asoc.2020.106705

[3] V.T. Hoang, M.D. Phung, T.H. Dinh, Q.P. Ha, System architecture for real-time surface inspection using multiple UAVs, IEEE Syst. J. 14 (2) (2020) 2925–2936. https://doi.org/10.1109/JSYST.2019.2922290

[4] C.H. Quach, M.D. Phung, H.V. Le, S. Perry, SupSLAM: a robust visual inertial SLAM system using superpoint for unmanned aerial vehicles, in: 2021 8th NAFOSTED Conference on Information and Computer Science (NICS), 2021, pp. 507–512. https://doi.org/10.1109/NICS54270.2021.9701527

[5] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, J.J. Leonard, Past, present, and future of simultaneous localization and mapping: towards the robust-perception age, IEEE Trans. Rob. 32 (6) (2016) 1309–1332. https://doi.org/10.1109/TRO.2016.2624754

[6] R. Mur-Artal, J.D. Tardos, ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras, IEEE Trans. Rob. 33 (5) (2017) 1255–1262. https://doi.org/10.1109/TRO.2017.2705103

[7] R. Li, Y. Lou, W. Song, Y. Wang, Z. Tu, Experimental evaluation of monocular visual inertial SLAM methods for freight railways, IEEE Sens. J. PP (Xx) (2023) 1. https://doi.org/10.1109/JSEN.2023.3301039

[8] W. Wang, Y. Hu, S. Scherer, TartanVO: a generalizable learning-based VO, Conf. Robot Learn. (2020). http://arxiv.org/abs/2011.00359.

[9] K. Wang, S. Ma, J. Chen, F. Ren, J. Lu, Approaches, challenges, and applications for deep visual odometry: toward complicated and emerging areas, IEEE Trans. Cognit. Dev. Syst. 14 (1) (2022) 35–49. https://doi.org/10.1109/TCDS.2020.3038898

[10] P.E. Sarlin, D. Detone, T. Malisiewicz, A. Rabinovich, SuperGlue: learning feature matching with graph neural networks, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (2020) 4937–4946. https://doi.org/10.1109/CVPR42600.2020.00499

[11] J. Ma, X. Jiang, A. Fan, J. Jiang, J. Yan, Image matching from handcrafted to deep features: a survey, Int. J. Comput. Vis. 129 (1) (2021) 23–79. https://doi.org/10.1007/s11263-020-01359-2

[12] T. Whelan, R.F. Salas-Moreno, B. Glocker, A.J. Davison, S. Leutenegger, ElasticFusion: real-time dense SLAM and light source estimation, Int. J. Rob. Res. 35 (14) (2016) 1697–1716. https://doi.org/10.1177/0278364916669237

[13] M. Labbé, F. Michaud, RTAB-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation, J. Field Rob. 36 (2) (2019) 416–446. https://doi.org/10.1002/rob.21831

[14] C. Li, S. Jiang, K. Zhou, DYR-SLAM: enhanced dynamic visual SLAM with YOLOv8 and RTAB-Map, J. Supercomput. 81 (5) (2025) 718.

[15] J. Delmerico, D. Scaramuzza, A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 2502–2509. https://doi.org/10.1109/ICRA.2018.8460664

[16] J. Han, R. Dong, J. Kan, BASL-AD SLAM: a robust deep-learning feature-based visual slam system with adaptive motion model, IEEE Trans. Intell. Transp. Syst. 25 (9) (2024) 11794–11804.

[17] H. Pu, J. Luo, G. Wang, T. Huang, H. Liu, Visual SLAM integration with semantic segmentation and deep learning: a review, IEEE Sens. J. 23 (19) (2023) 22119–22138.

[18] Y. Wang, Y. Tian, J. Chen, K. Xu, X. Ding, A survey of visual SLAM in dynamic environment: the evolution from geometric to semantic approaches, IEEE Trans. Instrum. Meas. 73 (2024) 1–21.

[19] W. Wang, D. Zhu, X. Wang, Y. Hu, Y. Qiu, C. Wang, Y. Hu, A. Kapoor, S. Scherer, TartanAir: a dataset to push the limits of visual SLAM, IEEE Int. Conf. Intell. Robots Syst. (2020) 4909–4916. https://doi.org/10.1109/IROS45743.2020.9341801

[20] D. Detone, T. Malisiewicz, A. Rabinovich, SuperPoint: self-supervised interest point detection and description, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 337–349. https://doi.org/10.1109/CVPRW.2018.00060

[21] J.A. Placed, J. Strader, H. Carrillo, N. Atanasov, V. Indelman, L. Carlone, J.A. Castellanos, A survey on active simultaneous localization and mapping: state of the art and new frontiers, IEEE Trans. Rob. 39 (3) (2023) 1686–1705.

[22] D. Scaramuzza, F. Fraundorfer, Visual odometry Part II, IEEE Rob. Autom. Mag. 19 (2) (2012) 78–90. https://doi.org/10.1109/MRA.2011.943233

[23] W. He, Z. Lu, X. Liu, Z. Xu, J. Zhang, C. Yang, L. Geng, A real-time and high precision hardware implementation of RANSAC algorithm for visual SLAM achieving mismatched feature point pair elimination, IEEE Trans. Circuits Syst. I Regul. Pap. 71 (11) (2024) 5102–5114.

[24] C. Campos, R. Elvira, J.J.G. Rodriguez, J.M.M. Montiel, D.T. Juan, ORB-SLAM3: an accurate open-source library for visual, visual-inertial, and multimap SLAM, IEEE Trans. Rob. 37 (6) (2021) 1874–1890. https://doi.org/10.1109/TRO.2021.3075644

[25] S. Wen, S. Tao, X. Liu, A. Babiarz, F.R. Yu, CD-SLAM: A real-time stereo visual–inertial SLAM for complex dynamic environments with semantic and geometric information, IEEE Trans. Instrum. Meas. 73 (2024) 1–8.

[26] D. Scaramuzza, Z. Zhang, Visual-inertial odometry of aerial robots, in: Encyclopedia of Robotics, Springer, 2019, (2019) 1–13.

[27] G. Huang, Visual-inertial navigation: a concise review, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 9572–9582. https://doi.org/10.1109/ICRA.2019.8793604

[28] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M.W. Achtelik, R. Siegwart, The EuRoC micro aerial vehicle datasets, Int. J. Rob. Res. 35 (10) (2016) 1157–1163. https://doi.org/10.1177/0278364915620033

[29] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, D. Cremers, J. Stuckler, D. Cremers, The TUM VI benchmark for evaluating visual-Inertial odometry, IEEE Int. Conf. Intell. Robots Syst. (2018) 1680–1687. https://doi.org/10.1109/IROS.2018.8593419

[30] A.L. Majdik, C. Till, D. Scaramuzza, The Zurich urban micro aerial vehicle dataset, Int. J. Rob. Res. 36 (3) (2017) 269–273. https://doi.org/10.1177/0278364917702237

[31] Z. Zhang, S. Liu, G. Tsai, H. Hu, C.C. Chu, F. Zheng, PIRVS: an advanced visual-inertial SLAM system with flexible sensor fusion and hardware co-design, Proc. - IEEE Int. Conf. Rob. Autom. (2018) 3826–3832. https://doi.org/10.1109/ICRA.2018.8460672

[32] J. Jeon, S. Jung, E. Lee, D. Choi, H. Myung, Run your visual-inertial odometry on NVIDIA jetson: benchmark tests on a micro aerial vehicle, IEEE Rob. Autom. Lett. 6 (3) (2021) 5332–5339. https://doi.org/10.1109/LRA.2021.3075141

[33] L. Zhang, M. Helmberger, L.F.T. Fu, D. Wisth, M. Camurri, D. Scaramuzza, M. Fallon, Hilti-oxford dataset: a millimeter-accurate benchmark for simultaneous localization and mapping, IEEE Robot. Autom. Lett. 8 (1) (2023) 408–415. https://doi.org/10.1109/LRA.2022.3226077

[34] S. Shah, D. Dey, C. Lovett, A. Kapoor, AirSim: high-fidelity visual and physical simulation for autonomous vehicles, (5) 2017. http://arxiv.org/abs/1705.05065.

[35] Y. Song, S. Naji, E. Kaufmann, A. Loquercio, D. Scaramuzza, FlightMare: a flexible quadrotor simulator, in: Proceedings of the 2020 Conference on Robot Learning vol. 155, 2020, pp. 1147–1157. https://proceedings.mlr.press/v155/song21a.{html} http://arxiv.org/abs/2009.00563.

[36] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014. arXiv preprint arXiv:1409.1556

[37] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, G. Huang, OpenVINS: a research platform for visual-inertial estimation, in: 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2020, pp. 4666–4672. https://doi.org/10.1109/ICRA40945.2020.9196524

[38] P. Lindenberger, P.-E. Sarlin, M. Pollefeys, LightGlue: local feature matching at light speed, in: ICCV, 2023.

[39] C. Campos, R. Elvira, J.J.G. Rodríguez, J.M.M. Montiel, J.D. Tardós, ORB-SLAM3: an accurate open-source library for visual, visual–inertial, and multimap SLAM, IEEE Trans. Rob. 37 (6) (2021) 1874–1890.

[40] S.L. Bowman, N. Atanasov, K. Daniilidis, G.J. Pappas, Probabilistic data association for semantic SLAM, in: Proceedings - IEEE International Conference on Robotics and Automation vol. 11 (8), 2017, pp. 1722–1729. https://doi.org/10.1109/ICRA.2017.7989203

[41] N. Atanasov, S.L. Bowman, K. Daniilidis, G.J. Pappas, A unifying view of geometry, semantics, and data association in SLAM, in: IJCAI International Joint Conference on Artificial Intelligence, 2018, pp. 5204–5208. https://doi.org/10.24963/ijcai.2018/722

[42] V. Sze, Y.-H. Chen, T.-J. Yang, J.S. Emer, Efficient processing of deep neural networks: a tutorial and survey, Proc. IEEE 105 (12) (2017) 2295–2329. https://doi.org/10.1109/JPROC.2017.2761740