2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

In-Flight Initialization of Global Visual-Inertial Estimators using Geospatial Data

Chunyu Li, Mengfan He, Xu Lyu and Ziyang Meng

***Abstract*— In this work, we propose a solution that leverages geospatial data to initialize the monocular visual-inertial navigation system. For Visual-Inertial Navigation Systems (VINS) operating on UAVs, the ability to perform initialization and relocalization in mid-air is essential. However, degenerate motion can cause VINS to lose scale, making traditional initialization algorithms less reliable. To address this issue, we fuse geographic information in the initialization process, and utilize a learning-based feature matching algorithm to associate the information with inertial states. The proposed approach demonstrates adaptability to the degenerate motions of UAVs and significantly surpasses the estimation accuracy of conventional VINS initialization algorithms. Compared to methods that assist initialization by using a laser-range-finder (LRF), the proposed method solely relies on low-cost satellite imagery and elevation information. We evaluate the proposed approach on a large-scale UAV dataset, and compare with existing methods. The results demonstrate the superior effectiveness of the proposed method.**

I. Introduction

Navigation of Unmanned Aerial Vehicles (UAVs) primarily depends on the Global Positioning System (GPS), yet the localization results are degraded by obstructions, multipath effects, and electronic interference. The Visual-Inertial Navigation System (VINS) effectively addresses this challenge by fusing angular velocity and acceleration data from the Inertial Measurement Unit (IMU) with observations of visual features from a camera. This integration enables the estimation of a UAV's pose in six degrees of freedom (6 d.o.f). However, visual tracking is fragile during fast motion of UAVs. In addition, visual features are situated at a distance far from the UAV, which further exacerbates the difficulty in estimating their positions. The excessive accumulated drift is also detrimental for reliable estimation. All these issues make airborne VINS prone to failures. Therefore, in-flight initialization when the algorithm fails is particularly crucial for VINS on UAVs.

VINS initialization involves determining the system's initial states using visual and inertial data, including attitude, position, velocity, and the biases of gyroscopes and accelerometers. Since the scale of positions and velocities is unobservable using merely monocular visual measurements, traditional monocular VINS algorithms rely on inertial measurements to recover the correct scale [1], [2]. However, such an approach requires the UAV to undergo sufficient motion to provide the IMU with adequate excitation. Achieving this can be challenging during rectilinear trajectories or zero acceleration motion, which UAVs frequently encounter [3]. Given the matching between the viewpoint of downward-facing cameras on UAVs and the satellite imagery, we propose a method that utilizes geospatial data, i.e., longitude, latitude and elevation, to enhance the initialization of VINS. Specifically, through learning-based visual place recognition (VPR) and feature matching methods, the geospatial data is associated with the visual features in the onboard image. Then, the matched geospatial data is used to obtain fully observable states during flight, enhancing the accuracy and reliability of the initialization. The main contributions of this work could be summarized as:

- We propose a hierarchical framework for coarse-to-fine geospatial data association. The framework achieves efficient image retrieval based on global image descriptors extraction, followed by the utilization of a deep

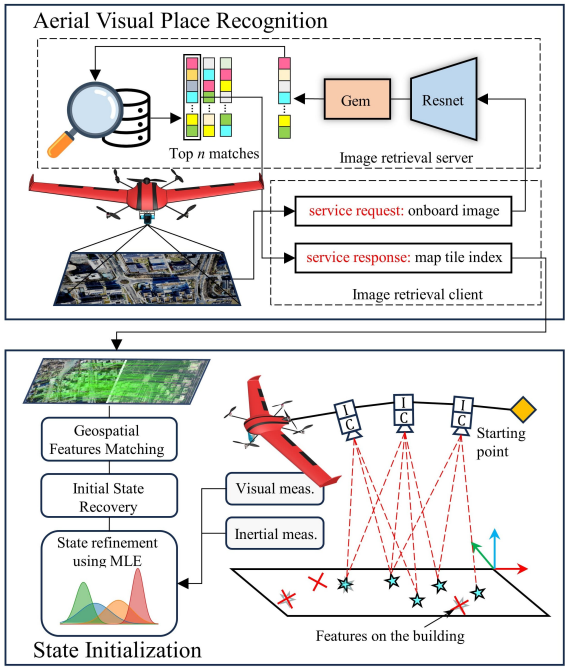


Fig. 1: Overview of the geospatial data-aided visual-inertial initialization.

*This work was supported by the National Natural Science Foundation of China under Grant 62273195, the National Key R&D Program of China under Grant 2022ZD0119601 and China Postdoctoral Science Foundation under Grant GZC20231302. (Corresponding author: Ziyang Meng.)

The authors are with Department of Precision Instrument, Tsinghua University, Beijing 100084, China (email: lcyfly1@163.com, hmf21@mails.tsinghua.edu.cn, lvclay@163.com, ziyangmeng@mail.tsinghua.edu.cn).

learning-based approach for local features matching for cross-domain images.
- Using the proposed solution, reliable initialization in degraded motion scenes can be achieved without the need for additional sensors. Also, the proposed scheme estimate complete initial states including 6 d.o.f poses, velocities, the bias vectors of inertial sensors and feature positions. Relying on these results, the subsequent estimations by VINS can be seamlessly employed.

II. Related Work

As shown in Fig. 1, the initialization task consists of two main components: aerial visual localization and inertial state estimation. For aerial visual localization, the most critical challenge is to overcome the differences in acquisition time, season, and perspective between satellite and airborne images to achieve their alignment. A neural network based method NetVLAD for extracting image global descriptors was proposed in [4], which learns to aggregate local descriptors. In [5] a segmentation network was used to extract scene information, enabling localization based solely on a downward-facing camera and satellite maps. Patel *et al.* [6] aligned aerial images with satellite images using a dense mutual information method, followed by estimating the UAV's pose using a Kalmen-filter algorithm. Then, Bianchi *et al.* [7] studied the storage and computational cost issues by training an encoder to compress images into a low-dimensional representation. Fragoso *et al.* [8] proposed a domain-adaptive image transformation network to transform images from different seasons to the same domain, thereby enabling matching and localization. Addressing the similar issue, Kinnari *et al.* [9], [10] trained an image similarity network that is invariant to seasons and used particle filtering for localization. Based on NetVLAD, a real-time pose estimation scheme was proposed for UAVs localization in [11], where satellite images and 3D reconstruction models were utilized to recover the pose via Perspective-n-Point (PnP). In general, the aforementioned works adopted a coarse-to-fine framework. This process begins with the extraction of a low-dimensional vector from the onboard image, which is then matched with a satellite database to identify the most similar image. Then a further localization is performed to achieve a more refined estimation. Inspired by these works, we also employ this framework, which allows us to efficiently associate geospatial data.

Given that VINS has 4 unobservable states, i.e., the 3D global position and yaw [12], its initializer can only assume the initial values in these directions to be zero, and then recover the rest variables. In [1], this problem was formulated to a quadratic constrained least-squares problem and a closed-form solution was provided. Qin *et al.* [2] proposed a robust initialization system, which aligns vision-only structure to the pre-integrated IMU measurements and recovers the metric scale, velocity, gravity vector, and gyroscope bias. The system was employed in VINS-Mono [13] as its initializer. To estimate states in the global frame, Qin *et al.* [14] utilized geographic data from GPS to realize globally consistent state estimation, where odometry results were only used to constrain relative states. A similar fusion framework was introduced in [15], which obtained 2D geographic data from satellite images rather than GPS. The estimators proposed in [16], [17] maintained the inertial states in the East-North-Up (ENU) frame. The relative transformation between the ENU and VINS frame was explicitly estimated, and then used to transform the states to the ENU frame. The aforementioned algorithms in [14]–[17] all required VINS to complete initialization first before estimating pose in the global frame. However, since the initialization process is short, and the UAVs usually undergo constant acceleration motions during this process, the traditional initialization tends to fail. In addition, for a UAV flying at altitudes over 150 meters, all features are usually distant from the camera, which results in larger depth estimation errors. Although it has been proven effective to use laser-range-finder (LRF) measurements in the initialization process [18], adding a sensor capable of measuring heights over 150 meters leads to an undesirable increase in the UAV's size, cost, weight, and power consumption. Moreover, since the aforementioned algorithms cannot provide initial states in the global frame, the relative transformation between the VINS and the global frames still needs to be estimated, which is complex and time-consuming. To tackle these issues, in this work, we leverage 3D geospatial data to directly provide reliable and accurate initial states for VINS in the ENU frame, establishing a foundation for a global navigation system that is independent of GPS.

III. System Overview

The overview of the system is illustrated in Fig. 1. We propose a coarse-to-fine localization algorithm that begins by performing aerial visual place recognition to achieve rough localization, that is, identifying the satellite map tile that most closely matches the onboard image. Next, we extract Superpoints [19] from both the onboard and corresponding satellite images, and use Superglue [20] to achieve feature matching between the cross-domain images, establishing associations between local features and latitude, longitude. The elevation data is then retrieved using the features' position from a digital elevation map (DEM). Utilizing the 3D-2D correspondences, a PnP is implemented to estimate the camera's poses, and a linear equation system is constructed based on IMU's pre-integration to recover the velocities. Finally, visual measurements, inertial measurements, and geographic information are formulated as a maximum likelihood estimation problem to optimize the initial states.

IV. Methodology

A. Offline Data Processing

As shown in Fig. 2, offline data processing includes preparing a satellite image database and a masked DEM. First, to achieve aerial visual place recognition, we crop the satellite image of the predetermined flight area into tiles, whose resolution is determined by the approximate flying height and the camera's field of view. Then we extract global

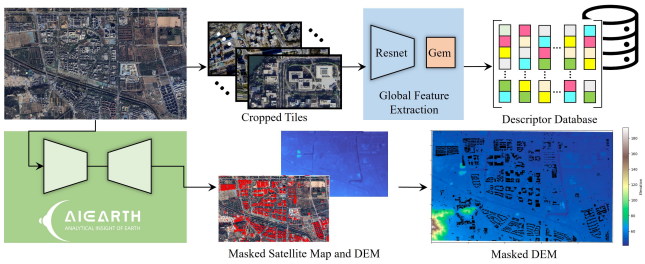


Fig. 2: Offline data preparation for satellite map database and masked DEM.

descriptors from each satellite tile using an encoder, which is described in Section IV-B, thereby forming a database. We use a DEM map to obtain the elevation information of the area. Given the typically inaccurate elevation information for buildings, we utilize the earth science cloud platform **AI Earth** to extract buildings from the entire satellite map and create a masked elevation map based on the extraction results, excluding the elevation information on buildings.

B. Aerial Visual Place Recognition

To achieve aerial visual place recognition, we employ a learning-based method to extract a discriminative descriptor of the image, and two steps are involved: feature extraction and feature aggregation. Specifically, ResNet [21] is chosen as the backbone to extract an $H \times W \times D$ feature map $\mathcal{M}$, where H and W denote the map's height and width respectively, and D is the length of the features' descriptors. Generalized Mean Pooling (GeM) [22] is then used to group the extracted features $\mathcal{M}$ and aggregate them to a global descriptor:

$$\mathcal{F} = \left(\sum_{i=1}^{H \times W} f_i^3 \right)^{\frac{1}{3}} \tag{1}$$

where $f_i \in \mathbb{R}^D$ is a per-pixel feature. The extracted global descriptor is retrieved in the aforementioned pre-built database, and we select the tile with the most similar descriptor as the matched image. Then a local features matching is performed between the matched tile and onboard image via Superpoints and SuperGlue. We consider the visual place recognition successful and proceed to the next step if there are enough features after the Random Sample Consensus (RANSAC) test.

C. Geospatial Features Matching

After retrieving the satellite tile image, local feature matching is performed to associate geospatial data to visual features. The matched features located on buildings are removed using the masked satellite map described in Section IV-A. The keyframe I_0 represents the first onboard image that successfully matched a satellite tile. Suppose that I_0 matches n local features to the satellite tile S_0, whose observation sets are $\mathcal{I}_0 := \{\mathbf{f}_0, \ldots, \mathbf{f}_{n-1}\}$ and $\mathcal{S}_0 := \{\mathbf{g}_0, \ldots, \mathbf{g}_{n-1}\}$ respectively. Then each feature's position ${}^E\mathbf{p}_{g_j}$ in the ENU frame can be calculated via the following mapping function:

$${}^E\mathbf{p}_{g_j} = \begin{bmatrix} x_j \\ y_j \\ z_j \end{bmatrix} = \begin{bmatrix} \mu\left(u_{g_j} - \frac{W}{2}\right) + x_E \\ -\mu\left(v_{g_j} - \frac{H}{2}\right) + y_N \\ h(\mathbf{g}_j) \end{bmatrix}, \forall j \in \{0, \ldots, n-1\} \tag{2}$$

where μ represents the resolution of the satellite image in meter per pixel, $\mathbf{g}_j = [u_{g_j}, v_{g_j}]^\top$ is the raw observation of the j-th feature in the image coordinate, H and W denote the width and height of the tile image respectively, x_E and y_E are the east and north coordinates of the tile's center in the ENU frame, and $h(\cdot)$ refers to the elevation looked up from the masked DEM map. It is worth noting that the extracted Superpoints are also tracked across consecutive frames, and the observations are utilized for state refinement in Section IV-E.

D. Initial States Recovery

The purpose of this step is to recover the initial states. Given all positions of the matched features and their observations, the camera poses $\{{}^{C_i}_{E}\mathbf{R}, {}^E\mathbf{p}_{C_i}\}, \forall i \in \{0, \ldots, k-1\}$ can be calculated using PnP, then the inertial rotation and position are calculated subsequently by:

$${}^{I_i}_{E}\mathbf{R} = {}^{C}_{I}\mathbf{R}^\top {}^{C_i}_{E}\mathbf{R} \tag{3}$$

$${}^E\mathbf{p}_{I_i} = {}^E\mathbf{p}_{C_i} + {}^{C_i}_{E}\mathbf{R}^\top {}^C\mathbf{p}_I \tag{4}$$

where ${}^{C}_{I}\mathbf{R}$ and ${}^C\mathbf{p}_I$ are calibrated camera-IMU extrinsics parameters. The position of the i-th frame can be also evolved from the initial timestamp via:

$${}^E\mathbf{p}_{I_i} = {}^E\mathbf{p}_{I_0} + {}^E\mathbf{v}_{I_0}\Delta t_i - \frac{1}{2}{}^E\mathbf{g}\Delta {t_i}^2 + {}^{I_0}_{E}\mathbf{R}^\top {}^0\boldsymbol{\alpha}_i \tag{5}$$

where $\Delta t_i = t_i - t_0$, ${}^E\mathbf{g}$ is the fixed gravity vector in the ENU frame. The position preintegration term ${}^0\boldsymbol{\alpha}_i$ from t_0 to t_i is defined by:

$${}^0\boldsymbol{\alpha}_i = \int_{t_0}^{t_i} \int_{t_0}^{s} {}^0_u\Delta\mathbf{R}\left(\mathbf{a}_m - \mathbf{b}_\mathbf{a} - \mathbf{n}_\mathbf{a}\right) duds \tag{6}$$

where $\mathbf{a}_m$ is the measured acceleration, $\mathbf{b}_\mathbf{a}$ and $\mathbf{n}_\mathbf{a}$ are the bias and noise vector respectively, and ${}^0_u\Delta\mathbf{R}$ is the relative rotation of the inertial frame from time u to t_0. Given (5), the velocity of the first keyframe can be expressed by:

$$\Delta t_i {}^E\mathbf{v}_{I_0} = {}^E\mathbf{p}_{I_i} - {}^E\mathbf{p}_{I_0} - \frac{1}{2}{}^E\mathbf{g}\Delta {t_i}^2 - {}^{I_0}_{E}\mathbf{R}^\top {}^0\boldsymbol{\alpha}_i. \tag{7}$$

We can then construct a linear equation system as:

$$\underbrace{\begin{bmatrix} \vdots \\ \Delta t_i \mathbf{I}_3 \\ \vdots \end{bmatrix}}_{\mathbf{A}} {}^E\mathbf{v}_{I_0} = \underbrace{\begin{bmatrix} \vdots \\ {}^E\mathbf{p}_{I_i} - {}^E\mathbf{p}_{I_0} - \frac{1}{2}{}^E\mathbf{g}\Delta {t_i}^2 - {}^{I_0}_{E}\mathbf{R}^\top {}^0\boldsymbol{\alpha}_i \\ \vdots \end{bmatrix}}_{\mathbf{b}}, \tag{8}$$

and calculate the initial velocity via:

$${}^E\mathbf{v}_{I_0} = \left(\mathbf{A}^\top\mathbf{A}\right)^{-1}\mathbf{A}^\top\mathbf{b}. \tag{9}$$

Then the rest velocities for $i \in \{1, \ldots, k-1\}$ can be recovered by:

$$ {}^{E}\mathbf{v}_{I_i} = {}^{E}\mathbf{v}_{I_0} - {}^{E}\mathbf{g}\Delta t_i + {}^{I_0}_{E}\mathbf{R}^{\top 0}\boldsymbol{\beta}_i \tag{10} $$

where the velocity preintegration ${}^{0}\boldsymbol{\beta}_i$ from t_0 to t_i is defined by:

$$ {}^{0}\boldsymbol{\beta}_i = \int_{t_0}^{t_i} {}^{0}_{u}\Delta\mathbf{R}\left(\mathbf{a}_m - \mathbf{b}_\mathbf{a} - \mathbf{n}_\mathbf{a}\right) du. \tag{11} $$

The piecewise constant measurements model proposed in [23] is used to calculate the position preintegration term ${}^{0}\boldsymbol{\alpha}_i$ and velocity preintegration term ${}^{0}\boldsymbol{\beta}_i$. The rough estimates for all matched features' positions are directly taken from the results in Section IV-C. Note that we assume the biases are both zero in this stage, and leave them to be refined in the next stage.

E. Maximum Likelihood Estimation

We follow the similar approach of OpenVINS initializer [24] for refining the state, which includes the following components:

$$ \mathbf{x} = \begin{bmatrix} \mathbf{x}_{I_0}^\top & \cdots & \mathbf{x}_{I_{k-1}}^\top & {}^{E}\mathbf{p}_{g_0}^\top & \cdots & {}^{E}\mathbf{p}_{g_{n-1}}^\top \end{bmatrix}^\top \tag{12} $$

$$ \mathbf{x}_{I_i} = \begin{bmatrix} {}^{I_i}_{E}\bar{q}^\top & {}^{E}\mathbf{p}_{I_i}^\top & {}^{E}\mathbf{v}_{I_i}^\top & \mathbf{b}_{g,i}^\top & \mathbf{b}_{a,i}^\top \end{bmatrix} \tag{13} $$

where ${}^{I_i}_{E}\bar{q}$ represents the unit quaternion defining the rotation $\mathbf{R}({}^{I_i}_{E}\bar{q}) = {}^{I_i}_{E}\mathbf{R}$, the initial values of $\mathbf{b}_{a,i}$ and $\mathbf{b}_{g,i}$ are assumed to be zero, and there are totally k keyframes and n global features. The maximum likelihood estimation is then formulated as a nonlinear least-square problem:

$$ \min_{\mathbf{x}} \Big\{ \sum \|\mathbf{0} - r_I(\mathbf{x})\|^2_{\mathbf{P}^{-1}} + \sum \|\mathbf{z}_c - h_c(\mathbf{x})\|^2_{\mathbf{R}_c^{-1}} + \sum \left\| {}^{E}\mathbf{p}_g - {}^{E}\hat{\mathbf{p}}_g \right\|^2_{\mathbf{R}_g^{-1}} \Big\} \tag{14} $$

where $r_I(\mathbf{x})$ represents the residual of inertial measurements, which is used to constrain the relative states of consecutive keyframes using orientation, position and velocities calculated in Section IV-D and preintegration terms. The measurement $\mathbf{z}_c$ is raw observations of a feature, and $h_c(\mathbf{x})$ is the projection function of the camera model. Readers are referred to [25] for more details. Given that the states have become fully observable with the features' global information, we have removed the prior cost associated with unobservable directions in the original OpenVINS initializer [24] and introduced a positional cost term for the visual landmarks. As shown in (2), the position prior ${}^{E}\mathbf{p}_g$ is derived using the geospatial data, and ${}^{E}\hat{\mathbf{p}}_g$ denotes the estimated positions in the state vector $\mathbf{x}$.

F. Global State Estimation

After the initial states recovery step, the states can be continuously estimated under the framework of multi-state constraint Kalman filter (MSCKF). To eliminate the position and yaw drift, we include the positions of geospatial landmarks in the state. The geospatial landmarks are tracked using Superpoint and SuperGlue across frames. Considering the learning-based feature matching is time-consuming, FAST corners [26] are still detected and tracked using the optical flow method [27] in another thread to constrain consecutive states. The state vector of our geospatial data aided visual-inertial system includes current inertial state $\mathbf{x}_I$, a set of c historical poses $\mathbf{x}_C$, a set of m standard SLAM features $\mathbf{x}_F$ [28], a set of n geo-referenced features $\mathbf{x}_G$:

$$ \mathbf{x}_i = \begin{bmatrix} \mathbf{x}_I^\top & \mathbf{x}_C^\top & \mathbf{x}_F^\top & \mathbf{x}_G^\top \end{bmatrix}^\top \tag{15} $$

$$ \mathbf{x}_I = \begin{bmatrix} {}^{I_i}_{E}\bar{q}^\top & {}^{E}\mathbf{p}_{I_i}^\top & {}^{E}\mathbf{v}_{I_i}^\top & \mathbf{b}_{\omega_i}^\top & \mathbf{b}_{a_i}^\top \end{bmatrix}^\top \tag{16} $$

$$ \mathbf{x}_C = \begin{bmatrix} {}^{I_{i-1}}_{E}\bar{q}^\top & {}^{E}\mathbf{p}_{I_{i-1}}^\top & \cdots & {}^{I_{i-c}}_{E}\bar{q}^\top & {}^{E}\mathbf{p}_{I_{i-c}}^\top \end{bmatrix}^\top \tag{17} $$

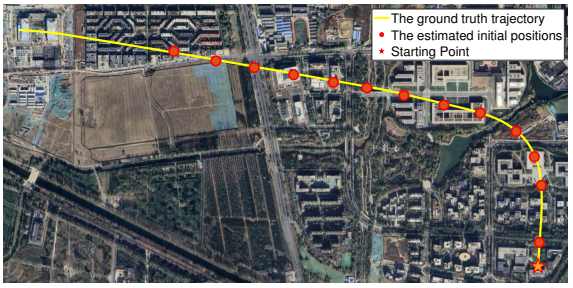
$$ \mathbf{x}_F = \begin{bmatrix} {}^{E}\mathbf{p}_{f_0}^\top & \cdots & {}^{E}\mathbf{p}_{f_{m-1}}^\top \end{bmatrix}^\top \tag{18} $$

$$ \mathbf{x}_G = \begin{bmatrix} {}^{E}\mathbf{p}_{g_0}^\top & \cdots & {}^{E}\mathbf{p}_{g_{n-1}}^\top \end{bmatrix}^\top. \tag{19} $$

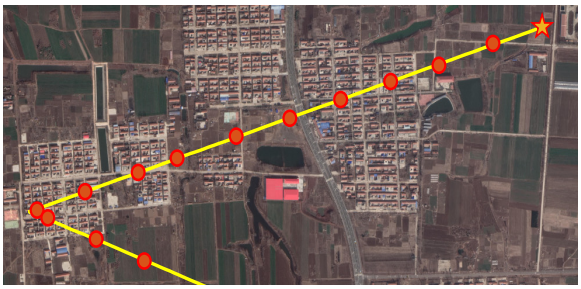
Besides the update using visual observations, a Kalman filter update is performed for the j-th geospatial feature in the state using the position measurement in (2):

$$ \begin{bmatrix} x_j \\ y_j \\ z_j \end{bmatrix} := \mathbf{z}_{g_j} = {}^{E}\mathbf{p}_{g_j} + \mathbf{n}_{g_j} \tag{20} $$

where the noise vector $\mathbf{n}_{g_j}$ is assumed to be zero-mean white Gaussian.



(a) Haidian Flight Area



(b) Jimo Flight Area

Fig. 3: The flight areas, ground truth trajectories, and initial positions of the two datasets

V. Experiments

A. Experimental Setup

To comprehensively validate the robustness of the algorithm, we include two flight trajectories in our experiments: one in a rural area and the other in an urban area. The trajectory in the rural area is acquired from our collected datasets, where a UAV flies over the rural region of Jimo,

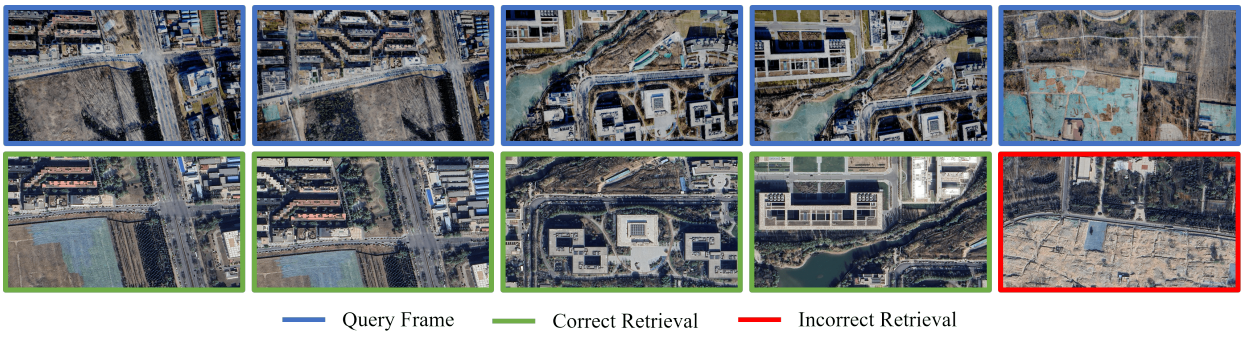
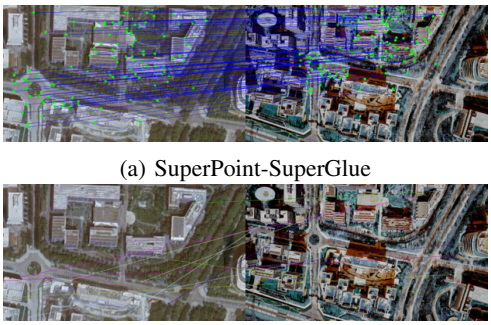


Fig. 4: Image retrieval examples.



(a) SuperPoint-SuperGlue

(b) SIFT-NNM

Fig. 5: Comparison of different feature matching methods. NNM here denotes the nearest neighbor matching method.

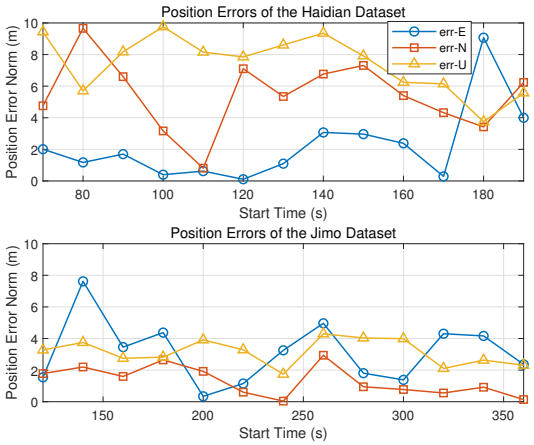


Fig. 6: Position errors in different directions, where 'err-E(N/U)' denotes the error in the east (north/up) direction.

Tsingdao [29]. The UAV is equipped with a downward-facing camera, rigidly attached to the body frame (IMU frame), and also featured a downward laser range finder, aligned parallel to the optical axis of the camera. During the flight, image data, inertial data, ranging data, as well as GPS coordinates and altitude are collected.

Considering the convenience of obtaining ground truth, we collect the dataset of urban scenes in a simulation environment. We first use aerial images to perform offline 3D reconstruction of an area in Haidian District, Beijing, and then import it into Airsim [30]. Next, we set up the UAV model and sensor models, and conduct an online flight in the simulation environment, collecting timestamps, noisy sensor data, and ground truth data during the flight. The sensor configuration mirrors that of the Jimo Flight, with the additional collection of global pose ground truth for subsequent comparisons.

In Fig. 3, aerial views of the two test areas are presented. For the Jimo flight, the UAV flies at an altitude of approximately 150 meters with a speed of around 8 m/s. The UAV in the Haidian flight, on the other hand, flies northwest at an altitude of 300 meters. The airspeed of the UAV was set between 16 to 20 m/s. The flight lengths for the rural and urban trajectories were set to 1800 meters and 3000 meters, respectively.

B. Geospatial Data Association

In the aerial visual place recognition process, we use the output feature of the 4th convolutional block of ResNet101 network. Additionally, we fine-tune the GeM aggregation layer with a self-built training dataset, which contains a large number of satellite images from different years. For the Haidian flight, the satellite map database consist of 1248 tiles and the entire area of selected map is about 13.63 square kilometers. We adopt the service mechanism of the Robot Operating System (ROS) to achieve aerial VPR. In the initialization node, the image retrieval client requests the VPR service every 0.8 seconds. Upon receiving a request, the retrieval server executes the VPR algorithm and returns the index of the most similar tile. Following this, the RANSAC test described in Section IV-B is then performed. If this retrieval is successful, the matched features are tracked for later calculation; otherwise, the process is repeated with the latest frame. The retrieval achieves 79.9% recall rate at top 1, which means most frames can find the correct satellite tile in the first time, and the recall rate reaches 94.8% at top 5. Some examples are shown in Fig. 4. In regions characterized by roads and buildings, the model is capable of guaranteeing successful retrieval. Conversely, in areas lacking texture, the performance of retrieval tends to deteriorate.

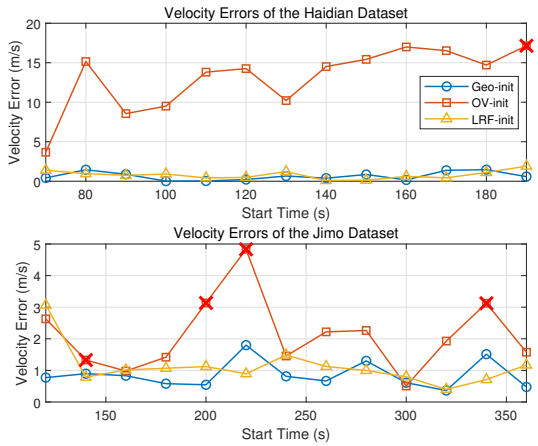


Fig. 7: Velocity estimated values and errors of different algorithms at different time instants. The cross here indicates that the state diverged in the subsequent estimates.

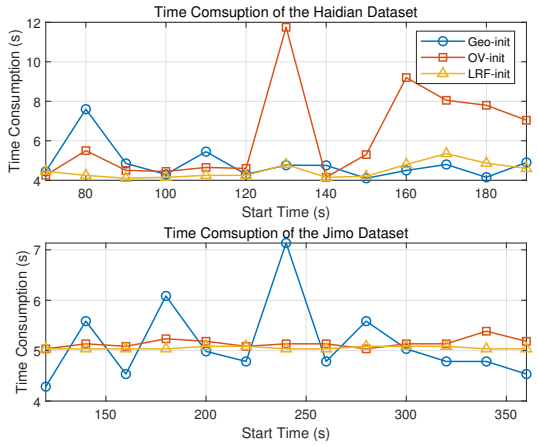


Fig. 8: Time consumption of different algorithms.

Learning-based approaches have been demonstrated to exhibit superior robustness to variations in acquisition time, lighting, environmental and perspectives changes when compared to traditional handcrafted feature matching methods [19], [20]. As shown in Figure 5, the combination of Superpoint and Superglue is effective in matching local features between the satellite tile and the onboard image, whereas the traditional SIFT method struggles with such cross-domain matching. In order to achieve faster feature matching speed, we accelerated the model using TensorRT based on [31]. The image resolution is 960×540, and 1024 Superpoints are extracted from each frame. Speeds of 10 fps and 5 fps are achieved on an NVIDIA-1660ti graphics card for image retrieval and local feature tracking, respectively.

C. Initialization Results

The estimated initial positions at different initialization moments are presented in Fig. 3. It is shown that the algorithm's estimated positions are distributed near the true trajectory, with a root-mean-square error of 5.85 meters

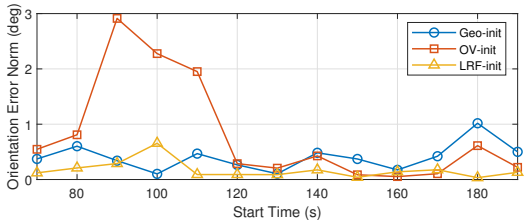


Fig. 9: Orientation errors of different algorithms at various initial moments during the Haidian Flight.

and 2.79 meters respectively for the urban and rural areas. Additionally, the position errors in the east, north, and up directions are depicted in Fig. 6. Notably, the error in altitude of the Haidin dataset is larger than those in the other two directions, a discrepancy that can be attributed to the lower accuracy of the DEM. We use the open source DEM from ALOS PALSAR with a resolution of 12.5 meters. To facilitate comprehensive comparison, we define the following variations of different initialization algorithms:

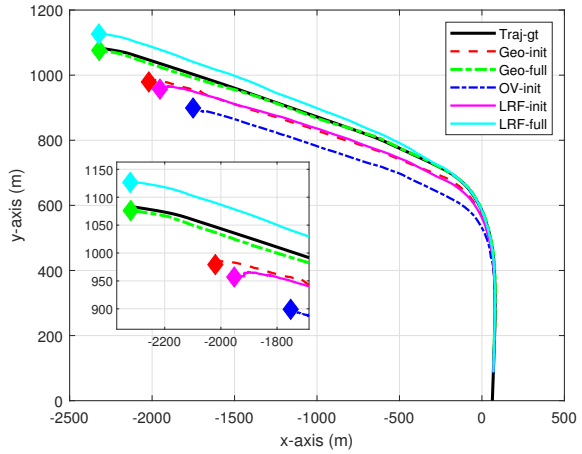
- **Geo-init**: Using geospatial information only during the initialization process.
- **Geo-full**: Continuous integration of geospatial information during the estimation process, as introduced in Section IV-F.
- **OV-init**: The original initialization algorithm of OpenVINS.
- **LRF-init**: Using range measurements only at the time of initialization.
- **LRF-full**: Using range measurements both during the initialization and state update processes [3].

To evaluate the effectiveness of different initialization algorithms in estimating velocity, the velocity errors at different starting times are presented in Fig. 7. It can be seen that the velocity errors of the original OpenVINS dynamic initializer are noticeably larger, which can be attributed to the unobservable scale problem. The results of the range-aided initialization algorithm, referred to as **LRF-init**, are also included in Fig. 7. The Root Mean Square Error (RMSE) for this method is slightly larger compared to our method, with values of 0.95 m/s versus 0.83 m/s for the urban flight, and 1.28 m/s versus 0.95 m/s for the rural flight, respectively. This comparison indicates that leveraging geospatial information enables achieving an observable scale without using additional sensors.

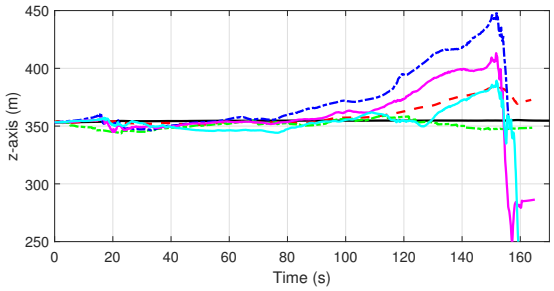
The comparisons of the time consumption of different algorithms are shown in Fig. 8. For the Haidian flight, we set an initialization time of 4 seconds and used 5 keyframes for optimization with both **OV-init** and **LRF-init**. In contrast, for the Jimo flight, we set the initialization time to 5 seconds. Due to the difficulty of recovering the correct scale, a shorter duration makes **OV-init** difficult to converge. For the **Geo-init** algorithm, due to the additional time required for geospatial feature matching, we reduce the number of keyframes to achieve similar time consumption. The results show that the **OV-init** takes the longest time

TABLE I: Experimental Results at Different Start Time. Green is the best, while blue represents the second best, and ATE-* and Ori-* denote the absolute trajectory errors and orientation errors of different algorithms respectively.

Start Time	70s	80s	90s	100s	110s	120s	130s	140s	150s	160s	170s	180s	190s
duration	170.63	160.63	150.63	140.63	130.63	120.63	110.63	100.63	90.63	80.63	70.63	60.63	50.63
ATE-Geo-init	97.38	129.95	76.90	70.79	100.54	75.95	74.82	73.34	33.61	49.00	51.79	23.41	**12.58**
ATE-Geo-full	**10.13**	**11.43**	**9.90**	**8.95**	**10.59**	**10.08**	**11.77**	**9.26**	**11.79**	**8.37**	**6.24**	**12.28**	**8.98**
ATE-OV-init	305.06	569.24	353.01	415.52	536.40	492.15	249.04	389.33	349.13	377.96	231.78	211.23	**failed**
ATE-LRF-init	177.81	174.89	174.76	229.66	**85.07**	121.34	182.55	79.86	71.52	48.48	59.31	18.56	13.19
ATE-LRF-full	**41.03**	**55.15**	**44.73**	**41.27**	87.37	**36.61**	**16.58**	**38.49**	**19.92**	**20.47**	**31.11**	**13.78**	13.69
Ori-Geo-init	2.45	**1.61**	**2.37**	**2.44**	**2.01**	2.08	2.04	**2.37**	2.80	**1.22**	**1.71**	**1.68**	**1.08**
Ori-Geo-full	**1.12**	**0.75**	**1.12**	**0.83**	**1.13**	**0.92**	**1.11**	**1.04**	**1.20**	**1.02**	**0.74**	**0.85**	**1.15**
Ori-OV-init	3.48	4.01	5.13	11.85	5.50	3.98	1.85	2.99	4.35	18.95	2.98	2.40	**failed**
Ori-LRF-init	2.18	2.09	2.80	5.19	3.88	**1.41**	3.09	2.63	2.10	2.94	1.96	2.05	2.23
Ori-LRF-full	**2.12**	2.02	2.44	2.95	4.10	2.19	**1.05**	3.56	**1.19**	2.40	2.23	1.79	1.78



(a) Estimated Trajectories in the east-north plane, where 'traj-gt' denotes the ground truth trajectory, and the diamond marks indicate the estimated trajectories' endpoints.



(b) Estimated altitude in the z-axis

Fig. 10: Trajectories and altitude estimation results.

to successfully estimate the initial states. This is because traditional initialization algorithms struggle to correctly estimate the scale at certain moments, leading to difficulties in convergence for subsequent optimizations. Success can only be achieved through repeated initialization processes. In contrast, **LRF-init** has the shortest time consumption. The initialization method of **Geo-init** takes longer time at moments of failure in airborne image retrieval (i.e., at 80 seconds and 110 seconds of Haidian flight), but generally matches the performance of **LRF-init** at other time instants.

Additionally, we compared the orientation RMSE of different algorithms on the Haidian dataset, with the results shown in Fig. 9. Considering that the yaw of the other two algorithms is unobservable, we only evaluated pitch and roll here. From the results, it can be seen that the original **OV-init** algorithm has the highest angular error (1.21 degrees), the **LRF-init** algorithm performs the best (0.23 degrees), and the **Geo-init** algorithm is intermediate (0.46 degrees).

D. State Estimation Results

Following the state's initialization, we compare the performance of different algorithms in subsequent estimation. Table I presents the Absolute Trajectory Error (ATE) in meters and RMSE results of attitude estimation in degrees for different algorithms in the Haidian flight dataset. The estimated trajectories and altitudes starting from 70 seconds are presented in Fig. 10. It is evident that **LRF-init** and **Geo-init** are more accurate than **OV-init**, suggesting that initial state scale observability enhances subsequent state estimation performance. Benefitting from the higher estimation accuracy of velocity and attitude, **Geo-init** outperforms **LRF-init** in subsequent estimations. However, the errors of these two algorithms are still significant, as the scale and pose drift during the subsequent estimation process. The results for **LRF-full** show that maintaining scale observability through range measurements significantly reduces pose error compared to **LRF-init**. However, Fig. 10 reveals that **LRF-full**'s trajectory still experiences drift when compared to the ground truth. This occurs because the global 3D position and yaw remain unobservable, leading to accumulated errors in these dimensions. On the other hand, the errors of **Geo-full** remain within a specific range, thanks to the continuous fusion of geospatial information. For the trajectory with various lengths, the ATE and orientation RMSE are remained around 10 meters and approximately 1 degree respectively. Note that, as shown in Fig. 9, although the **LRF-init** algorithm has higher accuracy in estimating roll and pitch angles at the initial moments, its overall orientation error surpasses that of

the algorithm using geospatial data due to the unobservability in its yaw direction.

VI. Conclusion

We introduce a novel framework for monocular visual-inertial estimators to directly initialize their states in the global frame. Specificly, initial pose, velocity and feature positions under the ENU frame as well as biases of IMU, are provided for subsequent state estimation. Aerial visual place recognition on satellite images database enables coarse localization, while deep learning-based feature matching algorithms enable successful fine-matching, realizing the association of geospatial information with features. Initial guesses of the pose and velocity are obtained using PnP and solving a linear equation system, respectively, and subsequent maximum likelihood estimation further optimizes the estimated results. The proposed algorithm achieves 5.85 meters and 2.79 meters in the urban and rural areas respectively. Furthermore, the improved accuracy in velocity and attitude estimation helps enhance the accuracy of the subsequent nonlinear filter-based visual-inertial estimator. Feature matching in certain outdoor areas, such as mountainous and river regions, remains challenging and is a focus of future research.

References

[1] T.-C. Dong-Si and A. I. Mourikis, "Estimator initialization in vision-aided inertial navigation with unknown camera-imu calibration," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2012, pp. 1064–1071.

[2] T. Qin and S. Shen, "Robust initialization of monocular visual-inertial estimation on aerial robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4225–4232.

[3] J. Delaune, D. S. Bayard, and R. Brockers, "Range-visual-inertial odometry: Scale observability without excitation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2421–2428, 2021.

[4] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5297–5307.

[5] A. Nassar, K. Amer, R. ElHakim, and M. ElHelw, "A deep cnn-based framework for enhanced aerial imagery registration with applications to uav geolocalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR)*, 2018, pp. 1513–1523.

[6] B. Patel, T. D. Barfoot, and A. P. Schoellig, "Visual localization with google earth images for robust global pose estimation of uavs," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6491–6497.

[7] M. Bianchi and T. D. Barfoot, "Uav localization using autoencoded satellite images," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1761–1768, 2021.

[8] A. T. Fragoso, C. T. Lee, A. S. McCoy, and S.-J. Chung, "A seasonally invariant deep transform for visual terrain-relative navigation," *Science Robotics*, vol. 6, no. 55, p. eabf3320, 2021.

[9] J. Kinnari, F. Verdoja, and V. Kyrki, "Season-invariant gnss-denied visual localization for uavs," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 232–10 239, 2022.

[10] J. Kinnari, R. Renzulli, F. Verdoja, and V. Kyrki, "Lsvl: Large-scale season-invariant visual localization for uavs," *Robotics and Autonomous Systems*, vol. 168, p. 104497, 2023.

[11] S. Chen, X. Wu, M. W. Mueller, and K. Sreenath, "Real-time geo-localization using satellite imagery and topography for unmanned aerial vehicles," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 2275–2281.

[12] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Consistency analysis and improvement of vision-aided inertial navigation," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 158–176, 2013.

[13] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[14] T. Qin, S. Cao, J. Pan, and S. Shen, "A general optimization-based framework for global pose estimation with multiple sensors," *arXiv preprint arXiv:1901.03642*, 2019.

[15] M. Jun, Z. Lilian, H. Xiaofeng, Q. Hao, and H. Xiaoping, "A 2d georeferenced map aided visual-inertial system for precise uav localization," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 4455–4462.

[16] W. Lee, K. Eckenhoff, P. Geneva, and G. Huang, "Intermittent gps-aided vio: Online initialization and calibration," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 5724–5731.

[17] W. Lee, P. Geneva, Y. Yang, and G. Huang, "Tightly-coupled gnss-aided visual-inertial localization," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9484–9491.

[18] M. Scheiber, J. Delaune, S. Weiss, and R. Brockers, "Mid-air range-visual-inertial estimator initialization for micro air vehicles," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7613–7619.

[19] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR)*, 2018, pp. 224–236.

[20] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 4938–4947.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[22] F. Radenović, G. Tolias, and O. Chum, "Fine-tuning cnn image retrieval with no human annotation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1655–1668, 2018.

[23] K. Eckenhoff, P. Geneva, and G. Huang, "Closed-form preintegration methods for graph-based visual–inertial navigation," *The International Journal of Robotics Research*, vol. 38, no. 5, pp. 563–586, 2019.

[24] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and 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.

[25] P. Geneva and G. Huang, "Openvins state initialization: Details and derivations," Tech. Rep. RPNG-2022-INIT, University of Delaware, 2022, Tech. Rep., available online: `https://pgeneva.com/downloads/reports/tr_init.pdf`.

[26] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*. Springer, 2006, pp. 430–443.

[27] J.-Y. Bouguet *et al.*, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," *Intel corporation*, vol. 5, no. 1-10, p. 4, 2001.

[28] M. Li and A. I. Mourikis, "High-precision, consistent ekf-based visual-inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.

[29] M. He, C. Chen, J. Liu, C. Li, X. Lyu, G. Huang, and Z. Meng, "AerialVL: A Dataset, Baseline and Algorithm Framework for Aerial-based Visual Localization with Reference Map," to appear in *IEEE Robotics and Automation Letters* 2024.

[30] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics: Results of the 11th International Conference*. Springer, 2018, pp. 621–635.

[31] K. Xu, Y. Hao, S. Yuan, C. Wang, and L. Xie, "Airvo: An illumination-robust point-line visual odometry," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 3429–3436.