

# A Line-Based Unmanned Aerial Vehicle Localization Framework Using Mid-Wave Infrared Observations

Jungwoo Huh<sup>\*</sup>, Jaekyung Kim<sup>\*</sup>, Kyungjune Lee<sup>\*</sup>,  
Ingu Park<sup>\*\*</sup>, Junhyeong Bak<sup>\*\*</sup>, Byungjin Kang<sup>\*\*</sup>, Sanghoon Lee<sup>o</sup>

## ABSTRACT

With the growing demand for unmanned aerial vehicle (UAV) localization systems that remain robust under challenging weather and lighting conditions, mid-wave infrared (MWIR) imagery has emerged as a promising alternative to conventional visible light (VL) imagery. However, existing VL-based localization methods are not directly applicable to MWIR images due to their fundamentally different visual characteristics. In this paper, we propose a novel UAV localization framework specifically designed for MWIR imagery. Our approach begins by detecting line features from MWIR frames, which are then matched to a predefined 3D line map of geologically meaningful structures. To establish robust correspondences under significant viewpoint changes, we introduce a 2D-3D line matching model trained using a synthetic dataset generated via a novel data augmentation strategy. The UAV's pose is subsequently refined by minimizing the reprojection error between matched 2D and 3D lines. We validate our method using simulated MWIR flight sequences rendered from 3D model of the Sinjin Island. Unlike state-of-the-art VL-based baselines, which suffer from degraded performance in the MWIR domain, our method achieves more accurate and robust localization. In addition, the proposed framework runs in real-time, making it suitable for practical deployment in onboard UAV systems.

**Key Words** : UAV localization, Mid-wave Infrared, Line Detection, Line Matching, Data Augmentation, Pose Refinement

## 1. Introduction

The development of low-cost motors and gyroscopic sensors has led to the widespread production of unmanned aerial vehicles (UAVs), which have become essential tools for capturing aerial imagery<sup>[1,2]</sup>. A key requirement for such tasks is accurate UAV localization, which involves estimating the vehicle's pose by integrating various sensor inputs, including GPS data, IMU measurements, and visual features<sup>[3]</sup>.

While GPS and IMU provide initial pose estimates, visual features are commonly used to refine the UAV's position and orientation. Many localization methods rely on visual features extracted from visible light (VL) imagery<sup>[4,5]</sup>. However, VL sensors are highly sensitive to lighting conditions and have poor transmittance in adverse weather, limiting their effectiveness. To address these limitations, remote sensing imagery, particularly mid-wave infrared (MWIR), offers a promising alternative, as MWIR is

※ This work was supported by LIG Nex1 under grant RZ04EO-001, AI-based Automatic Target Detection/Tracking Algorithm Technology.

♦ First Author : Yonsei University, Department of Electrical and Electronic Engineering, gjwddn9@yonsei.ac.kr, 학생회원

o Corresponding Author : Yonsei University, Department of Electrical and Electronic Engineering, slee@yonsei.ac.kr, 종신회원

\* Yonsei University, Department of Electrical and Electronic Engineering, jkkproject@yonsei.ac.kr; naive2kj90@yonsei.ac.kr

\*\* LIG Nex1 Co., Ltd, Missile System Seeker R&D, ingu.park2@lignex1.com; junhyeong.bak@lignex1.com; byungjin.kang@lignex1.com, 정회원

논문번호 : 202507-154-A-RU, Received July 10, 2025; Revised August 20, 2025; Accepted August 21, 2025

less affected by lighting and weather conditions. Despite these advantages, UAV localization using MWIR imagery remains underexplored, and effective methodologies for applying localization techniques in this domain are not well established.

However, existing localization methods designed for VL images<sup>[4,5]</sup> cannot be directly applied to UAV localization using MWIR imagery. The primary reason is that MWIR images exhibit significantly different characteristics from VL images, leading to a mismatch in the behavior of conventional visual descriptors. As shown in Fig. 1(a), keypoints extracted using the SIFT algorithm<sup>[6]</sup> are much sparser in MWIR images due to their predominantly texture-less regions, whereas VL images yield a dense set of keypoint descriptors. In contrast, Fig. 1(b) depicts that line features detected by the EDLines algorithm<sup>[7]</sup> show comparable results in both MWIR and VL images. This is because line detection relies primarily on strong image gradients rather than complex local descriptors, making it more robust to the modality gap between VL and MWIR.

However, lines without associated descriptors are unsuitable for reliable feature matching, which is essential for UAV pose refinement. Most UAV localization methods rely on matching visual features detected in the current frame with those from previous frames. Since the UAV's viewpoint changes over time, descriptors that are invariant to geometric transformations, such as SIFT, are commonly used to represent visual features. These descriptors allow for robust comparison of features despite changes in viewpoint or scale. In contrast, measuring the similarity between lines is inherently ambiguous without descriptive representations<sup>[8,9]</sup>. As illustrated in Fig. 2, existing line distance metrics often result in mis-

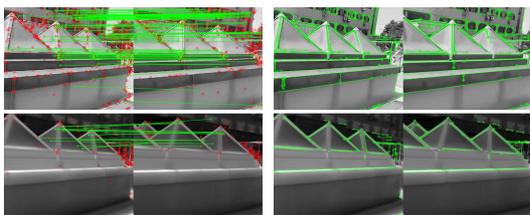


Fig. 1. Feature detection results between (a) SIFT descriptor and (b) EDLines.

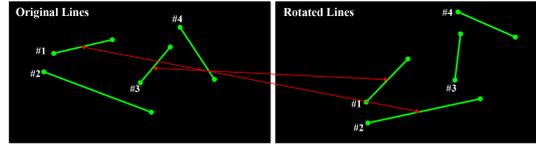


Fig. 2. Common mismatches between two set of lines using the angle distance of [9].

matches under significant rotation or translation. Therefore, for line-based localization to be viable, a more sophisticated matching strategy must be employed.

Motivated by these challenges, we propose a novel line-based UAV localization framework targeted for MWIR observations, which performs feature matching and pose refinement without relying on descriptors. An overview of the proposed framework is shown in Fig. 3. The UAV's initial pose, estimated using GPS and IMU measurements, is refined using visual cues from MWIR imagery. We adopt lines as the primary visual features, as line detection proves more reliable than keypoint detection in MWIR images. Given a predefined 3D line map and the initial pose estimate, 3D lines are transformed into the UAV's local coordinate system. These transformed lines are then matched with the detected 2D lines. Finally, the UAV pose is refined by minimizing the reprojection error between the 3D lines and their corresponding matched 2D observations.

To address the inherent ambiguity in line matching, we design a 2D-3D line matching model capable of establishing correspondences under large rotation and translation changes. Both 2D and 3D lines are represented using Plücker coordinates<sup>[10]</sup>, which encode

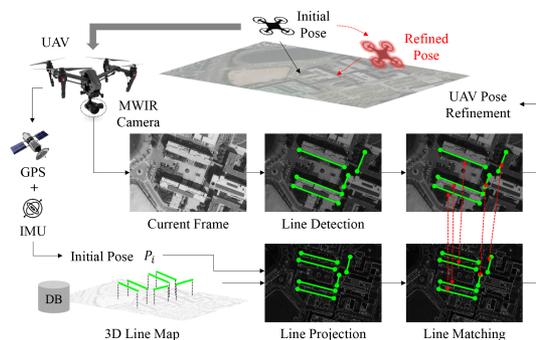


Fig. 3. Overview of the proposed MWIR imaging-based UAV localization method.

line geometry using direction and moment vectors rather than endpoints. This representation is significantly more stable, as endpoint-based coordinates are highly susceptible to detection noise, complicating model training. Our matching model learns the geometric structure of the 3D line map and the relationships between detected 2D lines through self- and cross-attention mechanisms<sup>[11]</sup>. The final line correspondences are determined by solving an optimal transport problem using the matching scores predicted by the model.

Training this matching model requires a sufficiently large dataset that captures the underlying geometry of the 3D map and the relationship to its 2D projections. However, collecting labeled data for a specific 3D line map is impractical, as it requires manual annotation of 2D-3D line correspondences. To overcome this limitation, we propose a novel data augmentation strategy that generates synthetic training data in a fully automated manner. Given a 3D line map, our method samples a reference UAV pose and a randomly perturbed pose to simulate realistic viewpoint changes. It then renders 2D projections of the 3D lines from the augmented pose and adds noise to the projected lines to mimic real-world 2D line detections. By training on this synthetic dataset, our model learns to generalize effectively to real-world MWIR scenarios without requiring manually labeled ground-truth matches.

Our main contributions are summarized as follows:

- We propose a novel UAV localization framework targeted for MWIR imagery, using line-based representations for enhanced robustness under challenging sensing conditions.
- We introduce a 2D-3D line matching model that is significantly more robust to large rotational and translational variations compared to conventional matching methods.
- We develop a data augmentation strategy that automatically generates realistic synthetic training data, enabling effective model training without the need for manually annotated real-world data.

## II. Methodology

### 2.1 Notations

Let  $\mathcal{L}_{3D} = \{L_i | L_i \in \mathbb{R}^{3 \times 2}, i = 1, \dots, M\}$  denote the set of predefined 3D lines, where each line  $L_i$  is represented by its two endpoints in world coordinates, typically corresponding to geological landmarks. Similarly, let  $\mathcal{L}_{2D} = \{l_i | l_i \in \mathbb{R}^{2 \times 2}, i = 1, \dots, N\}$  denote the set of 2D lines detected from the current MWIR image. The UAV's initial pose, estimated from sensor data, is denoted by  $P_i = [R_i / t_i]$ , where  $R_i \in \text{SO}(3)$  is the 3D rotation matrix and  $t_i \in \mathbb{R}^{3 \times 1}$  is the translation vector. The refined pose, obtained after line matching and pose optimization, is denoted by  $P_r = [R_r / t_r]$ , with analogous definitions for  $R_r$  and  $t_r$ . The transformed 3D line map  $\mathcal{L}'_{3D}$ , expressed in the UAV coordinate system defined by  $P_r$ , is given by:

$$\mathcal{L}'_{3D} = \{R_i L_i + T_i | L_i \in \mathcal{L}_{3D}, T_i = [t_i, t_i]\}. \quad (1)$$

### 2.2 Line Detection

As various line detection methods exist, it is important to select the approach most suitable for MWIR-based UAV localization. Broadly, line detection methods can be categorized into two groups: (1) image processing-based methods and (2) learning-based methods. We compare two widely used methods from each category: LSD<sup>[12]</sup> and EDLines<sup>[7]</sup> for image processing-based approaches, and AFM<sup>[13]</sup> and MLSD<sup>[14]</sup> for learning-based approaches. The comparison results are shown in Fig. 4.

As illustrated in Fig. 4(a) and Fig. 4(b), image processing-based methods such as LSD and ED-Lines reli-

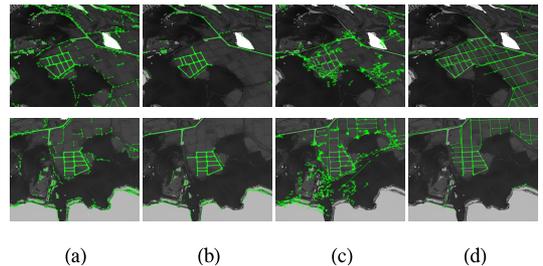


Fig. 4. Lines detected in MWIR images using (a) LSD, (b) EDLines, (c) AFM, and (d) M-LSD.

ably detect prominent lines in MWIR images by leveraging strong image gradients. In contrast, as shown in Fig. 4(c) and Fig. 4(d), learning-based methods such as AFM and M-LSD perform poorly in this domain, exhibiting significant noise and failing to extract meaningful lines. This degradation stems from the fact that these models are trained on VL images and thus learn feature distributions specific to the VL domain. Adapting them to MWIR imagery would require substantial retraining with annotated MWIR data, which is expensive and difficult to obtain. Given these observations, we conclude that image processing-based line detection methods are more suitable for MWIR-based UAV localization, as they require no training and are robust to domain shifts. Among them, we adopt EDLines over LSD, as it produces fewer but more salient lines, reducing clutter and improving subsequent line matching performance.

### 2.3 Line Matching

As previously shown in Fig. 2, conventional distance-based line matching methods<sup>[8,9]</sup> often result in mismatches, as the distance between two lines is inherently ambiguous without known correspondences. However, meaningful matches can still be inferred by leveraging the geometric relationships among neighboring lines. To exploit this relational information, we design a 2D-3D line matching model inspired by the graph attentional architecture<sup>[11]</sup>, as illustrated in Fig. 5. Our model begins by encoding the detected 2D lines  $L^{2D}$  and the transformed 3D lines  $L^{3D}$  using a line encoder and a midpoint encoder. To match the dimensionality between 2D and 3D lines, we represent the 2D lines in homogeneous coordinates. We then represent each line using Plücker coordinates<sup>[10]</sup>, which is composed of direction and moment vectors. Given a 3D  $\tilde{L}_i = [\tilde{x}_{1,i}, \tilde{x}_{2,i}]$  where  $\tilde{x}_{1,i}$  and  $\tilde{x}_{2,i}$  are its endpoints, the Plücker coordinates of  $\tilde{L}_i$  is computed as:

$$\begin{aligned} d &= \tilde{x}_{1,i} - \tilde{x}_{2,i}, \\ m &= \tilde{x}_{1,i} \times \tilde{x}_{2,i}. \end{aligned} \quad (2)$$

where  $d$  is the direction vector and  $m$  is the moment vector. The moment vector is the normal of the plane

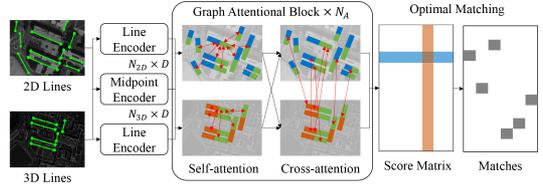


Fig. 5. Architecture of the proposed 2D-3D line matching model.

formed by the two endpoints and the origin.

For 2D lines, the direction vector is constrained by the image plane, and the moment vector  $m$  uniquely defines the line up to scale. Thus, we normalize the moment vector to obtain the final representation  $\bar{m} = m / \|m\|$ . Each 2D or 3D line is then encoded into a  $D$ -dimensional feature vector using a multi-layer perceptron (MLP), referred to as the line encoder. To differentiate between colinear lines, which cannot be distinguished solely by the moment vector, we also encode the midpoint  $\tilde{x}_{m,i} = (\tilde{x}_{1,i} + \tilde{x}_{2,i})/2$  of each line using a separate MLP, referred to as the midpoint encoder. The outputs of the line encoder and midpoint encoder are summed to produce the final line feature  $f_i \in \mathbb{R}^D$ , which is then passed to the graph attentional module.

The graph attentional module and matching mechanism follow the formulation of<sup>[11]</sup>. Given the encoded 2D and 3D line features  $F^{2D} = [f_1^{2D}, \dots, f_{N_{2D}}^{2D}]$  and  $F^{3D} = [f_1^{3D}, \dots, f_{N_{3D}}^{3D}]$ , the module computes both self-attention (within each set) and cross-attention (between sets), allowing contextual interactions among and across line features. This attentional block is repeated  $N_A$  times to progressively refine the features. To obtain the final matching, we first compute a score matrix which quantifies the similarity between refined 2D and 3D line features. The score matrix  $S \in \mathbb{R}^{(N_{3D}+1) \times (N_{2D}+1)}$  is computed using the inner-product between the refined features at the final layer  $L$ , augmented with a learnable dustbin score  $z$  for unmatched lines:

$$s_{i,j} = \begin{cases} f_i^{2D} \cdot f_j^{3D}, & i \in \mathcal{S}_{2D}, j \in \mathcal{S}_{3D}, \\ z, & i = N_{3D} + 1 \text{ or } j = N_{2D} + 1, \end{cases} \quad (3)$$

where  $\mathcal{S}_{3D} = \{1, 2, \dots, N_{3D}\}$  and  $\mathcal{S}_{2D} = \{1, 2, \dots, N_{2D}\}$ . To obtain the assignment, we solve an optimal transport problem over  $\mathcal{S}$  using the Sinkhorn algorithm<sup>[15]</sup>, which provides a differentiable approximation of the Hungarian algorithm<sup>[16]</sup>. This results in a soft assignment matrix  $\tilde{A}$ , which represents the model's confidence in each 2D-3D line pair.

The entire network is trained end-to-end using a loss function that supervises both matched and unmatched primitives:

$$L = - \sum_{(i,j) \in \hat{\mathcal{M}}} \log \tilde{A}_{i,j} - \sum_{i \in \mathcal{S}_{2D} \setminus \hat{\mathcal{M}}_i} \log \tilde{A}_{i, N_{3D}+1} - \sum_{j \in \mathcal{S}_{3D} \setminus \hat{\mathcal{M}}_j} \log \tilde{A}_{N_{2D}+1, j}, \quad (4)$$

where  $\hat{\mathcal{M}}$  denotes the set of ground-truth 2D-3D line correspondences, and  $\hat{\mathcal{M}}_i, \hat{\mathcal{M}}_j$  represent the sets of matched indices in 2D and 3D lines, respectively. The first term maximizes the confidence of correct matches, while the remaining terms guide the model to assign unmatched lines to the dustbin entry. Finally, the set of predicted matches is obtained by selecting, for each 3D line, the 2D line with the highest assignment score:

$$\mathcal{M} = \{(i, j) \mid i = \arg \max_{i' \in \mathcal{S}_{2D}} \tilde{A}_{i', j}, j \in \mathcal{S}_{3D}\}. \quad (5)$$

#### 2.4 Synthetic Data Augmentation

While careful model design is important, its effectiveness ultimately depends on the availability of appropriate training data. This is particularly critical for our 2D-3D line matching model, which must learn the geometric relationships specific to a given 3D line map. However, collecting such training data is labor-intensive, requiring MWIR image sequences, 2D line detection, and manual annotation of correspondences between 2D and 3D lines. To address this challenge, we propose a novel synthetic data augmentation method that generates realistic 2D-3D line pairs with minimal domain discrepancy, enabling effective model training without manual labeling. The overall augmentation process is shown in Fig. 6.

First, a point-of-focus (PoF)  $x_{\text{PoF}}$  is sampled near

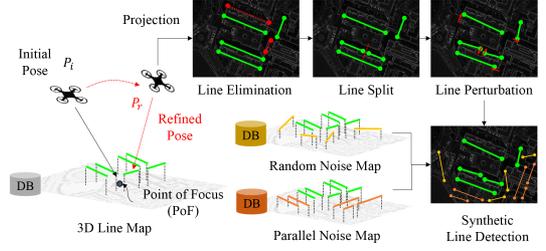


Fig. 6. Line matching data augmentation process.

the 3D line map  $\mathcal{L}_{3D}$ . This is achieved by drawing a 3D point from a Gaussian distribution defined by the mean and standard deviation of the midpoints of the lines in  $\mathcal{L}_{3D}$ . Next, we sample a random distance  $r \in [n, f]$ , azimuth angle  $\theta \in [0, 2\pi]$ , elevation angle  $\phi \in [0, \pi/2]$ , and twist angle  $\psi \in [-\pi/3, \pi/3]$ , where  $n$  and  $f$  denote the near and far bounds of the sampling radius. The initial UAV pose  $P_i$  is then defined as:

$$P_i = [R_i \mid t_i], \\ R_i = R_z(\psi)R_y(\phi)R_x(\theta), \quad (6) \\ t_i = x_{\text{PoF}} + rR_y(\phi)R_x(\theta)\hat{e}_x,$$

where  $R_x(\cdot), R_y(\cdot), R_z(\cdot)$  are rotation matrices around the respective axes, and  $\hat{e}_x$  is the unit vector along the x-axis. We can now obtain the transformed 3D line map  $\mathcal{L}_{3D}$  given the initial UAV pose using Eq.(1). The next step is to generate synthetic 2D lines to make pairs with the transformed 3D lines. We first transform the initial pose to a new pose  $P_r$ , which will be the target refined pose. The refined pose is derived by adding random rotation and translation as follows:

$$P_r = [R_r \mid t_r], \\ R_r = R_z(\delta\psi)R_y(\delta\phi)R_x(\delta\theta)R_i, \quad (7) \\ t_r = t_i + \delta r_x \hat{e}_x + \delta r_y \hat{e}_y + \delta r_z \hat{e}_z,$$

where  $\delta\psi, \delta\phi, \delta\theta, \delta r_x, \delta r_y, \delta r_z, \hat{e}_y, \hat{e}_z$  are random rotation angles and random translation distances with respect to  $x, y, z$  axes, and unit vectors in the  $y, z$ -axis direction, respectively.

Given the refined pose  $P_r$ , we project the 3D line map  $\mathcal{L}_{3D}$  to the  $P_r$  to obtain the projected 2D line map  $\mathcal{L}'_{2D} = \{l'_i \mid l'_i \in \mathbb{R}^{2 \times 2}, i = 1, \dots, N_{3D}\}$ . Each line in

$L_i = [x_{1,i}, x_{2,i}] \in \mathcal{L}_{3D}$  is projected by transforming its endpoints and applying the pinhole camera model:

$$l'_i = [\pi(K(R_r x_{1,i} + t_r)), \pi(K(R_r x_{2,i} + t_r))], \quad (8)$$

where  $K$  is the camera intrinsics and  $\pi(\cdot)$  denotes the perspective projection function:

$$\pi \left( \begin{bmatrix} x \\ y \\ z \end{bmatrix} \right) = \begin{bmatrix} x/z \\ y/z \end{bmatrix}. \quad (9)$$

From the projected 2D line map  $\mathcal{L}'_{2D}$ , we apply line noises which are frequently observed in real 2D line-detection as in Fig. 4. For each 2D line  $l'_{2D} \in \mathcal{L}'_{2D}$ , we randomly eliminate, split and perturb their endpoints.

After the target lines are augmented, additional noise lines are introduced to simulate realistic detection artifacts. Specifically, lines are sampled from two sources: a random noise map  $\mathcal{L}_R$  and a parallel noise map  $\mathcal{L}_P$ . The random noise map contains lines generated near the 3D line map  $\mathcal{L}_{3D}$ , while the parallel noise map includes lines that are spatially aligned and oriented parallel to lines in  $\mathcal{L}_{3D}$ . Using the same projection process applied to  $\tilde{\mathcal{L}}_{3D}$ , both  $\mathcal{L}_R$  and  $\mathcal{L}_P$  are projected to the refined pose  $P_r$  using Eq.(8), producing the projected sets  $\mathcal{L}'_R$  and  $\mathcal{L}'_P$ . These are then added to the augmented target lines  $\mathcal{L}'_{2D}$  to generate the final synthetic 2D line detection set  $\tilde{\mathcal{L}}_{2D}$ :

$$\tilde{\mathcal{L}}_{2D} = \mathcal{L}'_{2D} \cup \mathcal{L}'_R \cup \mathcal{L}'_P. \quad (10)$$

Since both the  $\tilde{\mathcal{L}}_{3D}$  and  $\tilde{\mathcal{L}}_{2D}$  are generated from the same underlying  $\mathcal{L}_{3D}$ , the ground-truth correspondences can be automatically derived without the need

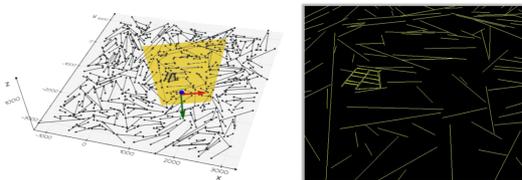


Fig. 7. Example visualization of the synthetic line detection generation process.

for manual labeling. An example generated synthetic line is shown in Fig. 7.

### 2.5 Pose Refinement

Finally, given the predicted matches  $\mathcal{M}$  between the detected 2D lines  $\tilde{\mathcal{L}}_{2D}$  and transformed 3D lines  $\tilde{\mathcal{L}}_{3D}$ , we can now estimate the refined pose  $P_r$  of the UAV from the initial pose  $P_i$ . Since the transformed lines  $\tilde{\mathcal{L}}_{3D}$  originate from the 3D line map  $\mathcal{L}_{3D}$ , their identities are preserved, allowing us to directly associate  $\tilde{\mathcal{L}}_{2D}$  with the corresponding lines in  $\mathcal{L}_{3D}$ . The refined pose  $P_r$  is obtained by minimizing the following energy function:

$$E(P_r) = \sum_{(i,j) \in \mathcal{M}} \|l'_i - \tilde{l}_j\|^2 + w \|m'_i - \tilde{m}_j\|^2 \quad (11)$$

where  $l'_i$  and  $m'_i$  represent the direction and moment vectors of the detected 2D line  $i$ , and  $\tilde{l}_j$  and  $\tilde{m}_j$  are the corresponding direction and moment vectors of the projected 3D line  $j$ . The scalar  $w$  is a weighting factor that controls the relative contribution of the moment term. It is evident from Eq.(8) that  $l'_i$  is a function of the refined pose  $P_r$ . Given the endpoints of  $l'_i$ , the corresponding moment vector  $m'_i$  can be computed using Eq.(2). Since  $l'_i$  and  $m'_i$  are differentiable with respect to  $P_r$ , the energy function  $E(P_r)$  can be efficiently minimized using gradient-based non-linear optimization methods.

### 2.6 Implementation Details

For the 2D-3D line matching model, we use a feature dimension of  $D = 128$  and apply a total of  $N_A = 3$  graph attention blocks. To generate the synthetic training dataset, we sample 10,000 distinct pairs of initial and refined UAV poses ( $P_i, P_r$ ) following the formulations in Eq.(6) and Eq.(7). The random distance  $r$  used to determine the initial pose is sampled uniformly from the range  $[n, f]$  with bounds  $n = 10$  and  $f = 200$ . Azimuth  $\theta$ , elevation  $\phi$ , and twist  $\psi$  angles are also sampled uniformly within their respective domains. For generating the refined pose  $P_r$ , random rotational perturbations  $\delta\psi, \delta\phi, \delta\theta$  are sampled from the range  $[-\pi/12, \pi/12]$ , and translational perturbations  $\delta r_x, \delta r_y, \delta r_z$  are sampled from  $[-10, 10]$ .

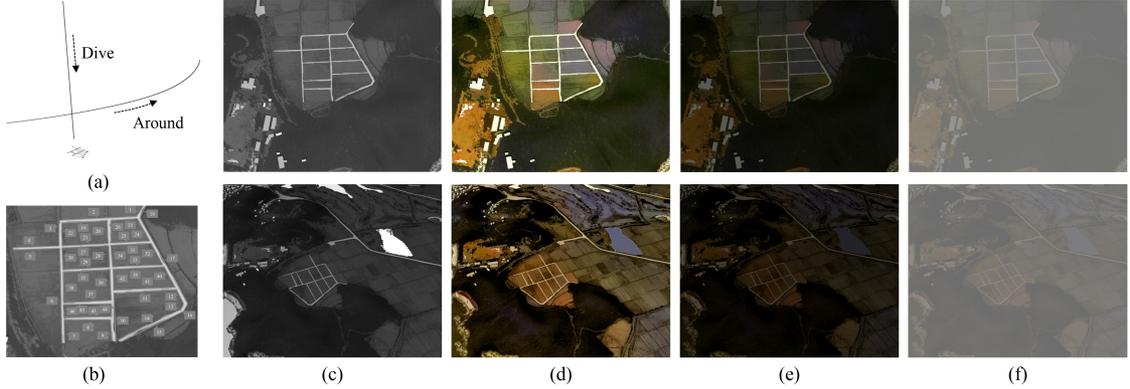


Fig. 8. Our simulated UAV flight dataset: (a) simulated flight sequences, (b) annotated 3D line map. Rendered images of Dive (Top) and Around (Bottom) sequences in (c) MWIR, (d) VL, (e) night time VL, (f) foggy VL.

To simulate realistic noise in 2D line detection, we generate 400 random and parallel lines around the 3D line map, ensuring no intersection with the existing lines. The 2D-3D line matching model is trained for 200 epochs using the Adam optimizer<sup>[17]</sup>, with a learning rate of  $5 \times 10^{-4}$  and a batch size of 64. For pose refinement, the weighting factor in the energy function is set to  $w = 10^{-5}$ , and the optimization is performed using the Levenberg-Marquardt algorithm<sup>[18]</sup>.

### III. Experiments

#### 3.1 Dataset

To evaluate the effectiveness of our MWIR-based UAV localization framework, we constructed a simulated UAV flight dataset using the physics-based sensor simulator provided by OKTAL-SE<sup>[19]</sup>, which is widely adopted for generating realistic optical imagery<sup>[20,21]</sup>. The simulation was performed over a detailed 3D model of Sinjin Island, with rendered images generated in both MWIR and VL bandwidth. We defined two distinct UAV flight patterns, Around and Dive, as illustrated in Fig. 8(a). These patterns are inspired by widely adopted flight configurations in benchmarks such as the EuRoC MAV Dataset<sup>[22]</sup>, which provide both easy and hard trajectories to evaluate robustness. In our dataset, the Dive sequence represents a relatively straightforward trajectory with limited viewpoint variation, whereas the Around sequence introduces significant angular and positional diversity, simulating more challenging conditions for

localization.

The sensors used in both simulations have a resolution of  $640 \times 512$  pixels. The intrinsic parameters are set to  $f_x = f_y = 1140$ , with a principal point at  $(p_x, p_y) = (320, 256)$ , while assuming negligible camera distortion. Each sequence lasts 15 seconds and is rendered at 30 frames per second, resulting in a total of 450 frames per sequence. All flight paths were simulated around the annotated 3D line map of Sinjin Island, which consists of geologically meaningful structural lines as shown in Fig. 8(b). To simulate realistic flight conditions, we added random rotational and translational perturbations to the smooth UAV trajectories, following the formulation in Eq. (7). During each simulation, both the UAV pose and the corresponding images were recorded for ground-truth reference. Importantly, our 2D-3D line matching model is trained exclusively on synthetic data generated from the 3D line map of Sinjin Island, without using any annotated MWIR or VL image sequences. Sample MWIR and VL renderings from the Dive and Around sequences are shown in Fig. 8(c) and (d), respectively.

To further validate the advantages of MWIR imagery over visible light (VL) under adverse conditions, we additionally rendered VL versions of both the Dive and Around sequences under two challenging scenarios. The first scenario simulates night-time conditions, where low-light settings were applied to reduce overall image brightness and contrast, as shown in Fig.8(e). The second scenario depicts foggy weather, in which visibility was degraded by overlaying a

semitransparent gray veil across the images, as illustrated in Fig.8(f). These conditions are designed to reflect realworld challenges where VL-based localization methods often fail, thereby emphasizing the robustness of MWIR-based localization.

### 3.2 Evaluation Metrics

To assess the performance of our MWIR-based UAV localization framework, we adopt standard evaluation metrics that are widely used in the visual localization and SLAM literature. Specifically, we evaluate translation and rotation error following established protocols in [4,5], and 2D-3D line matching precision, as commonly used in the feature matching literature<sup>[11]</sup>. The translation error  $E_t$  measures the accuracy of the estimated UAV position relative to the ground-truth trajectory. Given the predicted translation  $\hat{t}_i$  and ground-truth translation  $t_i$  at frame  $i$ , the error is defined as:

$$E_t = \frac{1}{N} \sum_{i=1}^N \|\hat{t}_i - t_i\|, \quad (12)$$

where  $N$  is the total number of frames, and  $\|\cdot\|$  denotes the Euclidean norm. The rotation error  $E_R$  quantifies the angular deviation between the estimated rotation matrix  $\hat{R}_i$  and the ground-truth rotation matrix  $R_i$ , and is computed as:

$$E_R = \frac{1}{N} \sum_{i=1}^N \arccos\left(\frac{\text{Tr}(R_i^T \hat{R}_i) - 1}{2}\right) \cdot \frac{180}{\pi}, \quad (13)$$

where  $\text{Tr}(\cdot)$  denotes the matrix trace. This yields the mean angular error in degrees across all frames. Finally, precision is used to evaluate the accuracy of the predicted 2D-3D line correspondences. A predicted match is considered a true positive (TP) if it correctly matches the ground-truth line pair, and a false positive (FP) otherwise. The precision is then defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (14)$$

which reflects the proportion of correct matches

among all predicted matches.

### 3.3 Quantitative Evaluation

#### 3.3.1 Effect of Visual Refinement

We first compared our method against baseline sensor configurations: raw GPS, raw IMU, GPS+IMU fusion, and our full pipeline, GPS+IMU with visual refinement. GPS is modeled as ground-truth positions corrupted with zero-mean Gaussian noise to approximate consumer-grade accuracy (20m mean position)<sup>[23]</sup>. Note that raw GPS provides position only.

The IMU baseline synthesizes gyro/accel signals with white noise and bias random walks and recovers state via strapdown mechanization<sup>[24]</sup>. The GPS+IMU base-line uses a loosely coupled EKF with position-only updates<sup>[25]</sup>, while our method further refines the fused pose via 2D-3D line correspondences. As shown in Table 1, GPS is noisy and provides no orientation, while IMU alone drifts and yields large translation errors. The GPS+IMU fusion approach reduces these errors but remains noise-limited. With using visual refinement, our method achieves the lowest translation and rotation errors, demonstrating the effectiveness of the proposed framework.

Table 1. Quantitative localization comparison between baseline sensor configurations.

Method	Dive		Around		Total	
	$E_t(\text{m})$	$E_R(^{\circ})$	$E_t(\text{m})$	$E_R(^{\circ})$	$E_t(\text{m})$	$E_R(^{\circ})$
GPS	15.6	-	21.3	-	18.5	-
IMU	88.0	13.8	51.4	9.0	69.7	11.4
GPS+IMU	57.0	13.8	25.6	9.0	41.3	11.4
Ours	<b>2.8</b>	<b>1.7</b>	<b>7.0</b>	<b>3.3</b>	<b>4.9</b>	<b>2.5</b>

#### 3.3.2 Comparison with VL-based Methods

We evaluated our method on both the Around and Dive sequences by measuring trajectory estimation accuracy and 2D-3D line matching precision. To assess the effectiveness of our MWIR-based localization framework, we compared it against two recent state-of-the-art methods developed for VL imagery, ORB-SLAM3<sup>[4]</sup> and DPVO<sup>[5]</sup>, on both VL and MWIR image sequences. ORB-SLAM3 is a classical simultaneous localization and mapping (SLAM) system that

Table 2. Quantitative localization comparison between VL-based methods on our simulated UAV flight sequences.

Method	Input	Dive					Around					Total				
		$E_t(m)$	$ER_t(^{\circ})$	TP	FP	Precision	$E_t(m)$	$ER_t(^{\circ})$	TP	FP	Precision	$E_t(m)$	$ER_t(^{\circ})$	TP	FP	Precision
ORB-SLAM3[4]	VL	8.6	2.1	-	-	-	15.8	6.2	-	-	-	12.2	4.1	-	-	-
	VL-Night	94.7	11.8	-	-	-	178.9	37.7	-	-	-	136.8	24.8	-	-	-
	VL_Foggy	59.3	9.1	-	-	-	138.4	29.4	-	-	-	98.9	19.3	-	-	-
	MWIR	54.2	3.6	-	-	-	87.4	26.6	-	-	-	70.8	15.1	-	-	-
DPVO[5]	VL	6.5	2.2	-	-	-	13.7	5.5	-	-	-	10.1	3.9	-	-	-
	VL-Night	79.1	9.8	-	-	-	158.3	31.1	-	-	-	118.7	20.5	-	-	-
	VL_Foggy	54.3	8.1	-	-	-	129.1	25.6	-	-	-	91.7	16.9	-	-	-
	MWIR	22.7	4.1	-	-	-	81.5	18.2	-	-	-	52.1	11.3	-	-	-
Ours	VL	2.7	1.6	823	233	0.779	6.6	3.1	903	113	0.889	4.7	2.4	1726	346	0.833
	VL-Night	75.2	9.0	437	555	0.441	150.3	29.0	482	516	0.483	112.7	19.6	919	1071	0.462
	VL_Foggy	52.8	7.8	517	596	0.464	120.5	24.0	561	579	0.492	86.6	16.1	1078	1175	0.479
	MWIR	<b>2.8</b>	<b>1.7</b>	798	266	0.750	<b>7.0</b>	<b>3.3</b>	881	129	0.872	<b>4.9</b>	<b>2.5</b>	1679	375	0.817

relies on handcrafted feature descriptors for keypoint detection and matching. DPVO is a deep learning-based visual odometry method that estimates camera motion by learning to match local image patches across frames for consistent frame-to-frame tracking. The trajectories for both ORB-SLAM3 and DPVO were estimated directly from the simulated image sequences. Since the estimated trajectories may differ from the groundtruth in scale, orientation, and translation, we aligned them using the Umeyama similarity transformation<sup>[26]</sup> to enable fair comparison. As both ORB-SLAM3 and DPVO do not perform 2D-3D line matching, we report only pose estimation results for these methods. The full comparison is summarized in Table 2.

As summarized in Table 2, the VL-based baselines behave as expected: they are reasonable on clean VL (e.g., ORB-SLAM3:  $E_t = 12.2$  m, DPVO:  $E_t = 10.1$  m) but degrade severely under VL-Night/Foggy (e.g., ORB-SLAM3 total  $E_t \approx 99$ -137 m; DPVO  $E_t \approx 92$ -119 m). When run on MWIR, both also deteriorate (ORB-SLAM3: 70.8 m, 15.1°; DPVO: 52.1 m, 11.3°), reflecting sparse keypoint detections in MWIR and modality issues due to VL-trained patch trackers. In contrast, our method maintains low error across modalities, 4.7 m/2.4° on VL and 4.9 m/2.5° on MWIR, and degrades more gracefully in VL-Night/Foggy (7.7-9.1 m, 3.5°-4.1°). These gains come from robust 2D-3D line matching, which is less sensitive to illumina-

tion and texture; the model achieves approximately 0.82 (82%) precision on MWIR overall (0.87 on Around), enabling accurate pose refinement.

### 3.4 Qualitative Evaluation

#### 3.4.1 Trajectory Visualization

We visualize the trajectories estimated by baseline sensor configurations and our framework in Fig.9(a) and (b). As expected, IMU-only suffers pronounced drift while GPS is reasonably accurate in position but does not provide orientation information. However, the loosely coupled GPS+IMU baseline still drifts as inertial errors accumulate, highlighting the limitation of localization in the absence of visual refinement. We also visualize the estimated trajectories also VL-based methods in Fig. 9(c) and (d). In Fig. 9(c), the Dive sequence involves less camera rotation than Around, resulting in relatively better performance for both ORB-SLAM3 and DPVO. As shown in Fig. 9(d), during the Around sequence, both ORB-SLAM3 and DPVO deviate significantly from the ground-truth trajectory from the beginning of the sequence. This degradation is primarily caused by the modality gap between VL and MWIR imagery, which leads to unreliable descriptor and patch matching. However, the estimated trajectories still exhibit instability, again due to poor descriptor and patch matching in the MWIR domain. In contrast, our method produces a more accurate and stable trajectory in both sequences, demon-

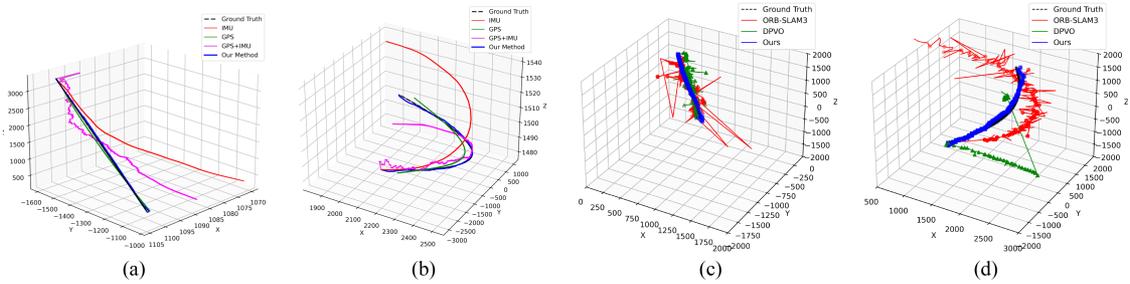


Fig. 9. Qualitative visualizations of trajectories. Comparison with baseline sensor configurations on (a) Dive and (b) Around sequences. Comparison between conventional methods using MWIR input on (a) Dive sequence (b) Around sequences.

strating its robustness and reliability under MWIR imaging conditions.

### 3.4.2 Matching and Refinement Visualization

We also visualize the 2D-3D line matching results between the transformed 3D line map  $\mathcal{L}_{3D}$  and the detected 2D lines  $\mathcal{L}_{2D}$ . Fig. 10(a) presents qualitative results for both the Around and Dive sequences. In both cases, our matching model successfully establishes correct correspondences, even under substantial viewpoint changes involving significant rotation and translation. Additionally, as shown in the rightmost matching visualization, the model remains robust to various line detection noises such as line elimination, splitting, and endpoint perturbation, which frequently occur when the UAV approaches the target lines. Despite these challenges, our model consistently identifies correct matches, demonstrating strong robust-

ness to real-world detection artifacts. To further assess the effectiveness of pose refinement, we visualize the projections of  $\mathcal{L}_{3D}$  before and after refinement in Fig. 10(b). In these visualizations, the red lines represent the projected 3D lines, while the green lines indicate the matched 2D detections. As shown in the second row of Fig. 10(b), the alignment between projected and detected lines significantly improves after pose refinement, confirming the accuracy of the estimated pose.

### 3.5 Ablation Studies

We performed ablation studies to investigate the impact of both the synthetic data augmentation strategy and key parameters of the 2D-3D line matching model. To evaluate the effectiveness of the augmentation strategy, we compared 2D-3D line matching accuracy under four settings: no augmentation, target

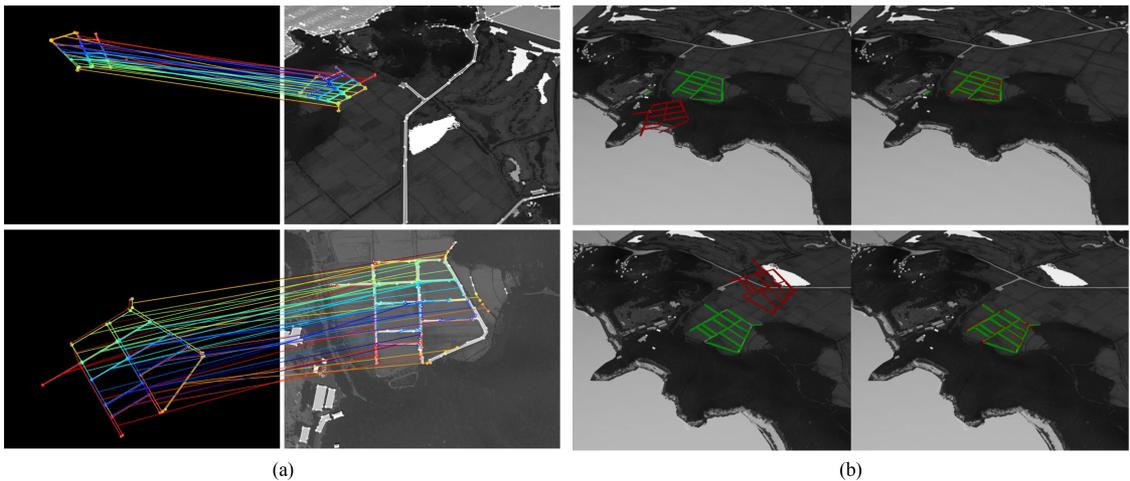


Fig. 10. Qualitative visualizations of (a) 2D-3D line matching and (b) reprojection of 3D lines after pose refinement.

line augmentation only, random line augmentation only, and both augmentations combined. The results, summarized in Table 3, show that without random line augmentation, the model achieves only 4-7% precision, whereas adding random line augmentation increases precision significantly, reaching up to 70%. Although target line augmentation alone does not significantly improve performance, it effectively reduces false positives. When combined with random augmentation, it improves precision by up to 11.35%, which is particularly beneficial as reducing false matches directly contributes to more accurate pose refinement.

We also searched for the optimal parameter settings of our matching model by varying the feature dimension size  $D$  and the number of attention blocks  $N_A$ . As shown in Table 4, the model achieved the highest matching precision when the feature dimension was set to  $D = 128$ . A lower feature dimension resulted in insufficient model capacity, while higher dimensions led to overfitting and degraded performance. Similarly, the best results for the number of attention blocks were observed at  $N_A = 3$ . Increasing the number of attention blocks beyond this did not yield further improvements and instead introduced more false positives, which negatively affects pose refinement.

Table 3. Ablation on data augmentation strategy.

Method	TP	FP	Precision
No Aug.	90	1921	0.0448
w/ Target Aug.	143	1824	0.0727
w/ Random Aug.	1480	618	0.7054
w/ Both	1679	375	0.8174

Table 4. Ablation on model parameters.

Model Parameters	TP	FP	Precision
$D=64$	1322	375	0.7790
$D=128$	1679	375	0.8174
$D=256$	1549	464	0.7695
$N_A = 1$	1189	387	0.7544
$N_A = 2$	1443	336	0.8111
$N_A = 3$	1679	375	0.8174
$N_A = 4$	1647	387	0.8097
$N_A = 5$	1650	393	0.8076
$N_A = 6$	1492	498	0.7497
$N_A = 7$	1517	414	0.7856
$N_A = 8$	1214	586	0.6744
$N_A = 9$	1538	430	0.7815

### 3.6 Runtime Evaluation

As efficient onboard processing is crucial for real-time UAV localization, we evaluate the computational performance of our proposed framework across two hardware platforms: a standard PC and an embedded system. The evaluation setup is shown in Fig. 11(a). The PC is equipped with an Intel Core i7-6700 CPU and an NVIDIA GeForce GTX 1050 Ti GPU. The embedded platform, shown in Fig. 11(b), is an NXP i.MX 8M Plus<sup>[27]</sup>, a low-power board suitable for UAV deployment. The software pipeline of our framework is illustrated in Fig. 11(c), where multiprocessing is employed to parallelize the key modules.

The overall runtime evaluation is conducted as follows. The 2D-3D line matching model is first initialized in the main process and converted to ONNX<sup>[28]</sup> format for lightweight and fast inference. The pipeline begins in the main process, simulating frame acquisition by reading MWIR image sequence frame-by-frame saved in a local directory. Three sub-processes are then spawned to handle the core tasks: Process #1 performs 2D line detection from the current MWIR frame, Process #2 conducts 2D-3D line matching using the ONNX model, and Process #3 performs pose refinement and visualization. The acquired frames are passed to Process #1, which performs 2D line detection. The detected lines are then sequentially forwarded to Process #2 for 2D-3D line matching, and subsequently to Process #3 for pose refinement and visualization, completing the full localization loop. Table 5 summarizes the runtime of each module and

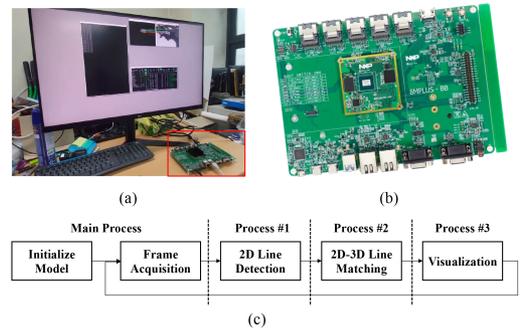


Fig. 11. Runtime analysis hardware and software settings. (a) Evaluation setup (b) NXP i.MX 8M Plus board (c) Software pipeline with multiprocessing.

Table 5. Runtime performance on PC and embedded board.

Hardware	Main Proc.	Proc. #1	Proc. #2	Proc. #3	FPS
PC	0.0121	0.0022	0.0154	0.0164	61.0
Embedded Board	0.0239	0.0213	0.0492	0.0475	20.3

the overall throughput. The framework runs at average 61 FPS on the standard PC and 20.3 FPS on the embedded board, confirming its real-time capability even on resource-constrained hardware.

#### IV. Conclusion

We proposed a novel UAV localization framework for MWIR imagery, addressing the limitations of VL-based methods under challenging visibility conditions. Our approach leverages 2D-3D line matching between MWIR-detected lines and a predefined 3D line map, enabled by a robust matching model trained entirely on a synthetically augmented dataset. The framework achieves high matching precision and accurate pose refinement, demonstrating strong performance compared to VL-based baselines on simulated MWIR flights. It also runs in real-time on both desktop and embedded platforms, supporting practical deployment. A current limitation is the reliance on a predefined 3D line map. In future work, we aim to extend our framework to handle unseen environments by generalizing the matching model for online or map-free localization.

#### References

- [1] U. Niethammer, M. James, S. Rothmund, J. Travelletti, and M. Joswig, "Uav-based remote sensing of the super-sauze landslide: Evaluation and results," *Eng. Geology*, vol. 128, pp. 2-11, Mar. 2012. (<https://doi.org/10.1016/j.enggeo.2011.03.012>)
- [2] S. H. Oh, S. W. Park, and J. G. Kim, "Uavbased coastal drowning detection method," *Korea Artificial Intell. Conf.*, pp. 129-130, Jeju Island, Korea, Sep. 2024.
- [3] J. Wendel, O. Meister, C. Schlaile, and G. F. Trommer, "An integrated GPS/MEMS-IMU navigation system for an autonomous helicopter," *Aerospace Sci. and Technol.*, vol. 10, no. 6, pp. 527-533, Sep. 2006. (<https://doi.org/10.1016/j.ast.2006.04.002>)
- [4] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam," *IEEE Trans. Robotics*, vol. 37, no. 6, pp. 1874-1890, Dec. 2021. (<https://doi.org/10.1109/TRO.2021.3075644>)
- [5] Z. Teed, L. Lipson, and J. Deng, "Deep patch visual odometry," *Advances in NeurIPS*, vol. 36, pp. 39033-39051, New Orleans, LA, USA, Dec. 2023.
- [6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Computer Vision*, vol. 60, no. 2, pp. 91-110, Nov. 2004. (<https://doi.org/10.1023/B:VISI.0000029664.99615.94>)
- [7] C. Akinlar and C. Topal, "Edlines: A real-time line segment detector with a false detection control," *Pattern Recognition Lett.*, vol. 32, no. 13, pp. 1633-1642, Oct. 2011. (<https://doi.org/10.1016/j.patrec.2011.06.001>)
- [8] H. Yu, W. Zhen, W. Yang, and S. Scherer, "Line-based 2-d-3-d registration and camera localization in structured environments," *IEEE Trans. Instrumentation and Measurement*, vol. 69, no. 11, pp. 8962-8972, Nov. 2020. (<https://doi.org/10.1109/TIM.2020.2999137>)
- [9] H. Yu, W. Zhen, W. Yang, J. Zhang, and S. Scherer, "Monocular camera localization in prior lidar maps with 2d-3d line correspondences," in *2020 IEEE/RSJ Int. Conf. IROS*, pp. 4588-4594, Las Vegas, NV, USA, Oct. 2020. (<https://doi.org/10.1109/IROS45743.2020.9341690>)
- [10] S. Teller and M. Hohmeyer, "Determining the lines through four lines," *J. Graphics Tools*, vol. 4, no. 3, pp. 11-22, Nov. 1999. (<https://doi.org/10.1080/10867651.1999.104875>)

- 06)
- [11] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proc. IEEE/CVF Conf. CVPR*, pp. 4938-4947, Seattle, WA, USA, Jun. 2020. (<https://doi.org/10.1109/CVPR42600.2020.00499>)
- [12] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: A line segment detector," *Image Process. On Line*, vol. 2, pp. 35-55, Mar. 2012. (<https://doi.org/10.5201/ipol.2012.gjmr-lsd>)
- [13] N. Xue, S. Bai, F. Wang, G.-S. Xia, T. Wu, and L. Zhang, "Learning attraction field representation for robust line segment detection," in *Proc. IEEE/CVF Conf. CVPR*, pp. 1595-1603, Long Beach, CA, USA, Jun. 2019. (<https://doi.org/10.5201/CVPR.2019.00169>)
- [14] G. Gu, B. Ko, S. Go, S.-H. Lee, J. Lee, and M. Shin, "Towards light-weight and real-time line segment detection," in *Proc. AAAI Conf. Artificial Intell.*, vol. 36, pp. 726-734, Virtual, Feb. 2022. (<https://doi.org/10.1609/aaai.v36i1.19953>)
- [15] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," *Advances in NeurIPS*, vol. 26, pp. 2292-2300, Lake Tahoe, NV, USA, Dec. 2013.
- [16] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Ind. and Appl. Math.*, vol. 5, no. 1, pp. 32-38, 1957. (<https://doi.org/10.1137/0105003>)
- [17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, San Diego, CA, USA, May 2015. (<https://doi.org/10.48550/arXiv.1412.6980>)
- [18] J. J. Moré, "The Levenberg-Marquardt algorithm: Implementation and theory," in *Numerical Analysis*, G. A. Watson, Ed., Lecture Notes in Mathematics, vol. 630, pp. 105-116, Aug. 2006. (<https://doi.org/10.1007/BFb0067700>)
- [19] OKTAL-SE, *Oktal-se-simulation environment for infrared and radar applications*, Retrieved Jun., 4, 2025, from <https://www.oktal-se.fr/>
- [20] S. Kim, W.-J. Song, and S.-H. Kim, "Robust ground target detection by sar and ir sensor fusion using adaboost-based feature selection," *Sensors*, vol. 16, no. 7, p. 1117, Jul. 2016. (<https://doi.org/10.3990/s160771117>)
- [21] T. Kim and H. Bang, "Fractal texture enhancement of simulated infrared images using a cnn-based neural style transfer algorithm with a histogram matching technique," *Sensors*, vol. 23, no. 1, p. 422, Dec. 2022. (<https://doi.org/10.3990/s23010422>)
- [22] M. Burri, J. Nikolic, P. Gohl, et al., "The euroc micro aerial vehicle datasets," *Int. J. Robotics Res.*, vol. 35, no. 10, pp. 1157-1163, Sep. 2016. (<https://doi.org/10.1177/0278364915620033>)
- [23] K. Merry and P. Bettinger, "Smartphone GPS accuracy study in an urban environment," *PloS one*, vol. 14, no. 7, e0219890, Jul. 2019. (<https://doi.org/10.1371/journal.pone.0219890>)
- [24] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technol.*, 2nd ed., The Institution of Eng. and Technol. (IET), 2004. (<https://doi.org/10.1049/PBRA017E>)
- [25] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Syst.*, 2nd ed., Artech House, 2013. (<https://doi.org/10.1017/S03734633313000672>)
- [26] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Trans. Pattern Anal. & Mach. Intell.*, vol. 13, no. 04, pp. 376-380, Apr. 1991. (<https://doi.org/10.1109/34.88573>)
- [27] NXP Semiconductors, *i.MX 8M Plus Evaluation Kit (8MPLUSLPD4-EVK)*, Retrieved Jun., 4., 2025, from <https://www.nxp.com/design/design-center/development-boards-and-designs/8MPLUSLPD4-EVK>
- [28] Microsoft Corporation, *ONNX Runtime: Accelerating Machine Learning Inferencing*, Retrieved Jun., 4., 2025, <https://onnxruntime.ai/>

### Jungwoo Huh



2017 : B.S. degree, Yonsei University

2017~Present : Integrated M.S & Ph.D. Program, Yonsei University

<Research Interest> Human Motion Estimation, Camera

Pose Estimation, Intelligent Surveillance

[ORCID:0000-0002-1103-8309]

### Ingu Park



2006 : B.S. degree, Pusan National University

2008 : M.S. degree, Pusan National University

2007~Present : Chief Research Engineer, Missile System Seeker R&D, LIG Nex1 Co., Ltd.

<Research Interest> Embedded System, Image Processing, Visual Object Detection and Tracking

[ORCID:0009-0009-1849-1067]

### Jaekyung Kim



2017 : B.S. degree, Yonsei University

2017~Present : Integrated M.S & Ph.D. Program, Yonsei University

<Research Interest> Light Simulation, Photo-realistic

Rendering, Object Detection

[ORCID:0000-0001-5142-8814]

### Junhyeong Bak



2020 : B.S. degree, Hannam University

2022 : M.S. degree, Inha University

2022~Present : Research Engineer, Missile System Seeker R&D, LIG Nex1 Co., Ltd.

<Research Interest> Fast Visual Object Detection and Tracking

[ORCID:0009-0009-4006-2357]

### Kyungjune Lee



2018 : B.S. degree, Yonsei University

2018~Present : Integrated M.S & Ph.D. Program, Yonsei University

<Research Interest> Camera Path Planning, Dynamic 3D

Registration, Multi-modal Content Generation

[ORCID:0009-0003-9309-5722]

### Byungjin Kang



2020 : B.S. degree, Yeungnam University

2022 : M.S. degree, Yeungnam University

2022~Present : Research Engineer, Missile System Seeker R&D, LIG Nex1 Co., Ltd.

<Research Interest> Infrared Image Object Detection and Tracking

[ORCID:0000-0002-1527-7884]

## Sanghoon Lee



1989 : B.S. degree, Yonsei University

1991 : M.S. degree, Korea Advanced Institute of Science and Technology (KAIST)

1991~1996 : Member of Technical Staff, Korea Telecom (KT)

2000 : Ph.D. degree, University of Texas at Austin

2000~2002 : Member of Technical Staff, Lucent Technologies

2003~2007 : Assistant Professor, Department of Electrical and Electronic Engineering, Yonsei University

2007~2012 : Associate Professor, Department of Electrical and Electronic Engineering, Yonsei University

2012~Present : Full Professor, Department of Electrical and Electronic Engineering, Yonsei University

<Research Interest> Quality Assessment, 3D Character Animation, Multi-modal Content Generation

[ORCID:0000-0001-9895-5347]