ARC-Loc: Leveraging Azimuthal Ray Convergence as a Geometric Cue for Direct Cross-View Localization

Hyeongsik Kim*[1], Mincheol Kim*[1], Heejoon Moon[2], and Je Hyeong Hong†[1,2,3]

[1] Dept. of Artificial Intelligence Semiconductor Engineering, Hanyang University, Republic of Korea
[2] Dept. of Artificial Intelligence, Hanyang University, Republic of Korea
[3] Dept. of Electronic Engineering, Hanyang University, Republic of Korea
khs06007,tlfj02,wilko97,jhh37@hanyang.ac.kr
*Equal contribution. †Corresponding author.
https://github.com/SpatialAILab/ARC-Loc

Abstract. Cross-view localization (CVL) estimates the pose of a ground image by matching it to a geo-referenced satellite image. To bridge the extreme viewpoint gap, mainstream pipelines rely on Bird's-Eye-View (BEV) transformations or 2D-to-3D lifting. However, deriving 3D structures from a single ground image is fundamentally ill-posed, causing these methods to endure geometric distortions and computational costs during 3D lifting or BEV projection. Furthermore, relying on external depth foundation models to resolve this introduces latency and remains susceptible to noisy predictions. In this work, we present a different approach inspired by a human navigation technique called *resection*, that can perform direct ground-to-satellite image matching and localization without relying on external depth foundation models. The key insights of our method are that (i) ground keypoints can be translated into azimuthal rays on the satellite map, and (ii) these rays ideally converge at the user location. Exploiting this geometric constraint through direct line-to-point correspondences, we introduce a minimal Azimuthal Ray Convergence (ARC) solver to identify the intersection alongside an ARC loss to optimize the matching network. By eliminating dependencies on computationally heavy BEV transformations and external depth foundation models, our approach achieves faster, memory-efficient inference, while its explicit feature matching ensures straightforward compatibility with existing frameworks. Experiments on VIGOR and KITTI demonstrate that ARC-Loc maintains competitive localization accuracy compared to recent approaches, highlighting its practicality.

Keywords: cross-view localization · azimuthal ray · resection · correspondence matching

1 Introduction

Cross-view localization (CVL) aims to estimate the pose of a ground camera given a geo-referenced aerial image. This is particularly crucial for autonomous

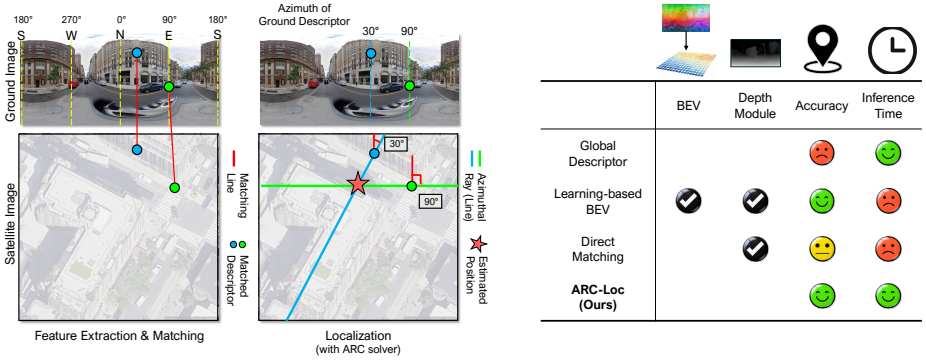

Fig. 1: *(Left)* **Localization procedure of ARC-Loc**, *(Right)* **comparison of CVL approaches**. Given the matched descriptors across the ground image and the aerial satellite image, each ground keypoint's azimuth is mapped into a line constraint on the satellite map by drawing an *azimuthal ray* from the corresponding satellite keypoint along the ground-view azimuth. The user device must lie on this ray, and the location is obtained by the intersection point of these rays. Based on this mechanism, ARC-Loc does not require computationally heavy BEV transformations or external depth foundation models, achieving faster inference times and competitive localization accuracy.

driving and urban planning in environments such as urban canyons, which are highly prone to GNSS denial and multipath interference [3]. While consumer-grade IMUs can reliably provide stable global orientation [5, 23], recovering the precise global position remains highly challenging. Existing HD map-based approaches that solve this demand massive costs for creation and maintenance [33]. In response, CVL has emerged as a promising alternative by matching ground images against globally accessible, geo-referenced satellite imagery. Nevertheless, bridging the visual gap caused by the drastic viewpoint differences between the two domains remains a formidable task.

As shown in Fig. 1, prior CVL approaches generally fall into three categories, each facing distinct challenges in bridging the severe ground-to-aerial viewpoint gap. Global descriptor-based methods [12, 29, 30] estimate pose by comparing single-vector representations, inherently sacrificing the fine-grained geometric details needed for precise localization. To preserve spatial cues, recent methods project ground features into an orthographic Bird's-Eye-View (BEV) space [6,10, 13, 25, 28]. However, projecting 2D ground pixels into a BEV coordinate system fundamentally requires unavailable 3D priors, introducing geometric distortion and additional computation for estimating depth, either via implicit learning [6, 25, 28] or by relying on depth foundation models [25, 28].

Furthermore, rendering a BEV view requires non-trivial height estimation to determine exactly which ground pixels correspond to surfaces actually visible from the satellite [26, 28], often leading to undesired geometric distortions. Alternatively, a direct matching-based framework such as Loc^2 [31] bypasses BEV warping through direct 2D-2D correspondences, yet still relies on external depth priors for 3D lifting [15], introducing sensitivity to depth inaccuracies. Thus, ex-

isting paradigms leave ample room for cross-view localization strategies free from geometric distortions, computational cost, and/or external depth dependencies.

A fundamental reason existing frameworks rely on complex 3D or BEV representations is their attempt to simultaneously resolve both translation and rotation. In the real-world applications such as autonomous vehicles [4, 8], the reliable global orientation can be obtained by standard IMUs [5, 23] or orientation prediction networks [19, 25]. Hence, we can reframe the cross-view localization task to focus on estimating precise position. For this purpose, we draw inspiration from a traditional human navigation technique called *resection* [1], in which an observer can determine his or her exact position by identifying visible landmarks, measuring the azimuth (bearing) to each landmark, and drawing corresponding rays on the 2D (aerial) map. The true location then lies at the convergence point of these azimuthal rays. We hypothesize that this purely 2D, line-intersection constraint can be directly adapted for cross-view localization, where ground-level visual features act as the azimuth measurements and the 2D aerial map serves as the global coordinate frame.

Motivated by the above principle, we present **ARC-Loc**, a novel framework that leverages Azimuthal Ray Convergence (ARC) as a geometric cue for cross-view localization. Specifically, we can interpret direct 2D ground-to-2D satellite point correspondences as 2D line (azimuthal rays)-to-2D point constraints (Fig. 1), and subsequently the camera position can be estimated at the convergence point of these azimuthal rays. To implement this, we introduce the *ARC solver*, a differentiable line-based solver capable of estimating the camera position from a minimal set of just two ground-aerial correspondences. Furthermore, to train the network for robust feature matching without dense correspondence labels, we propose the *ARC loss*, which enforces both accurate position estimation and the convergence of the mapped azimuthal rays. By operating entirely in the native 2D image domain without dependencies on external depth foundation models or BEV transformation, ARC-Loc avoids geometric distortions and achieves faster, memory-efficient inference. Moreover, ARC-Loc enables explicit, interpretable local feature matching similar to [28,31]. This provides straightforward compatibility with geometric correspondence-based pipelines [28,31], while seamlessly supporting RANSAC for robust cross-view localization.
To summarize, our key contributions are as follows:

- **ARC-Loc:** a cross-view localization framework inspired by a traditional navigation technique that eliminates dependency on BEV transformation or external depth models, achieving faster and memory-efficient inference.
- **ARC solver:** a differentiable (closed-form) minimal solver for estimating camera position from minimum of two azimuthal ray-to-aerial point constraints that is also compatible with RANSAC for robust localization.
- **ARC loss:** a loss function based on the geometric constraint of azimuthal ray convergence for effective training of feature matching network without requiring dense correspondence labels or depth priors.

Extensive experiments on VIGOR and KITTI datasets demonstrate that ARC-Loc achieves competitive localization accuracy while eliminating reliance on BEV

transformations and external depth foundation models, highlighting its practicability for real-world autonomous applications.

2 Related work

Global descriptor-based approaches. Early cross-view localization largely treated the task as image retrieval by learning a shared embedding for ground and satellite views [9, 24]. Subsequent works showed that these global representations can be extended to metric localization by regressing pose from global features [34]. To improve spatial precision within this paradigm, later methods introduced finer matching mechanisms such as grid-structured descriptors [30], cyclic/rolling matching in CCVPE to address orientation uncertainty [29], and cross-attention between ground features and satellite patches [12, 32]. Nevertheless, global-descriptor approaches inherently compress images into compact vectors, sacrificing geometric details, and their accuracy is also bounded by satellite patch resolution, making high-precision (*i.e.* sub-meter) localization challenging.

BEV-based approaches. To bridge the extreme viewpoint gap between perspective ground views and orthographic satellite views, the dominant trend has shifted towards transforming ground images into an intermediate Bird's-Eye-View (BEV) representation. Initial approaches, such as DenseFlow [22] and HC-Net [27], relied on explicit geometric projections like Inverse Perspective Mapping (IPM). While computationally efficient, IPM relies on the flat-world assumption, leading to geometric distortion or loss of vertical structures (*e.g.* buildings, trees), which are critical landmarks in urban environments [25]. To mitigate the geometric distortions, recent methods [6, 13, 20, 28] employ learning-based transformations. GGCVT [20] and Fevers *et al.* [6] utilize the transformer model to learn the perspective-to-BEV mapping implicitly. More recently, FG^2 [28] adapts the BEVFormer [13] architecture, lifting 2D image features into a 3D voxel space via deformable attention to synthesize a BEV feature map and performing 2D Procrustes alignment [17] to estimate pose. Similarly, BevSplat [25] leverages 3D Gaussian Splatting [10] to render high-quality BEV features, where a depth foundation model [15] is utilized to supervise the generation of 3D Gaussian primitives. While recent learning-based BEV methods are elevating the localization accuracy, performing accurate BEV warping remains fundamentally challenging due to the inherent lack of exact 3D priors. Furthermore, rendering a BEV view requires non-trivial height estimation to determine which ground pixels are actually visible from the satellite [25, 26]. Consequently, compensating for this missing 3D structure via explicit depth estimation or implicit BEV transformations [6, 25, 28] not only demands heavy additional computation, but inherently leads to geometric distortions during BEV warping.

Direct local feature matching-based approach. The most recent work, Loc^2 [31], takes a slightly different approach by matching directly in the original domain. They extract ground and aerial features from each respective image,

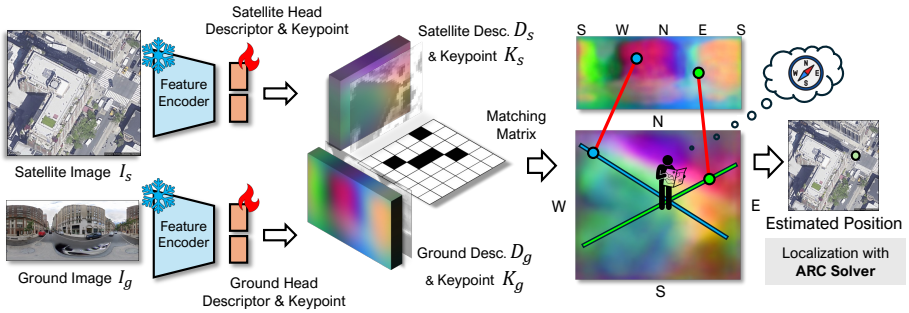

Fig. 2: Overview of ARC-Loc. The pipeline comprises two main stages: feature matching and localization. Initially, we extract dense features using a pretrained visual feature encoder and pass them through keypoint and descriptor heads. Then, by leveraging a matching matrix based on descriptor similarity and keypoint confidence, ARC-Loc establishes direct ground-aerial correspondences and transforms the ground matches into azimuthal rays. ARC solver then estimates the camera position by finding the intersection of these rays. During inference, as our solver requires a minimal set of just two correspondences, RANSAC can be equipped, enabling robust estimation.

lift the matched ground keypoints to the 3D space using a depth foundation model [15], and estimate pose via Procrustes alignment [17]. Through this explicit matching approach, they provide greater interpretability compared to previous implicit approaches, enable the use of RANSAC for robust estimation, and mitigate the burden of BEV transformations. Nevertheless, its reliance on an external depth foundation model not only introduces computational overhead but also makes the estimated pose highly sensitive to noisy depth predictions, resulting in slightly lower accuracy than recent CVL methods as shown in Tab. 1.

3 Proposed method

We now illustrate the overall pipeline of our ARC-Loc method. Following several previous studies in cross-view localization [19, 25], our framework assumes the heading prior is provided to some degree of accuracy. This is partly due to the fact that the ground heading can be readily obtained from consumer-grade IMUs [5, 33] (*e.g.* modern electronic devices such as cellphones, cars, etc.) or orientation prediction network [19]. Subsequently, our framework is primarily designed to estimate precise camera position under known orientation, although it remains robust to practical degree of orientation noise as shown in Fig. 4.

3.1 Framework overview

The proposed framework processes a ground image I_g and a geo-referenced satellite image I_s to estimate the camera's position $\mathbf{p} \in \mathbb{R}^2$. As shown in Fig. 2, the pipeline begins by extracting dense features to perform local feature matching, establishing direct 2D-2D cross-view correspondences between ground keypoints

$\{\mathbf{u}_i \in \mathbb{R}^2\}$ and satellite keypoints $\{\mathbf{x}_i \in \mathbb{R}^2\}$. Each pair is then assigned with a matching confidence weight w_i, where i denotes the correspondence index. Then, the matched ground keypoints are converted to respective azimuthal rays on the satellite image, providing geometric cues. Finally, we estimate the camera position $\mathbf{p}$ using our proposed ARC solver.

3.2 Cross-view feature extraction and direct matching

Feature extraction. Following prior cross-view localization works that leverage vision foundation models [25, 28, 31], we adopt the pretrained RADIOv3-L [16] as the backbone feature encoder. Given the cross-view images I_g and I_s, the backbone extracts dense feature maps $\mathbf{F}_g \in \mathbb{R}^{C\times H_g\times W_g}$ and $\mathbf{F}_s \in \mathbb{R}^{C\times H_s\times W_s}$, respectively, where C denotes the output channel dimension. Since these models are known to provide rich semantic understanding and discriminative local representations, we anticipate these help to bridge the extreme visual gap between ground and satellite views.

The obtained features from the above encoder are then passed to two lightweight heads of identical architecture, namely a *descriptor head* (f_D) and a *keypoint head* (f_K) both of which extract necessary keypoint information for matching correspondences. Both heads essentially consist of multiple convolutional layers followed by a self-attention layer to integrate contextual information. Since we have separate heads for the ground view (f_D^g and f_K^g) and satellite view (f_D^s and f_K^s), we yield two dense descriptor maps ($\mathbf{D}_g$, $\mathbf{D}_s$) and two keypoint confidence maps ($\mathbf{K}_g$, $\mathbf{K}_s$) defined as follows:

$$\begin{aligned} \mathbf{D}_g &= f_D^g(\mathbf{F}_g) \in \mathbb{R}^{\frac{C}{8}\times H_g\times W_g}, \quad & \mathbf{D}_s &= f_D^s(\mathbf{F}_s) \in \mathbb{R}^{\frac{C}{8}\times H_s\times W_s}, \\ \mathbf{K}_g &= f_K^g(\mathbf{F}_g) \in \mathbb{R}^{H_g\times W_g}, & \mathbf{K}_s &= f_K^s(\mathbf{F}_s) \in \mathbb{R}^{H_s\times W_s}. \end{aligned} \tag{1}$$

Direct feature matching. To compute the matching probabilities between cross-view image descriptors, we first define the pairwise similarity score as $Sim_{mn} = \mathrm{sim}(\mathbf{d}_g^m, \mathbf{d}_s^n)/\tau$, where $\mathrm{sim}(\cdot,\cdot)$ denotes the cosine similarity. Here, $\mathbf{d}_g^m$ and $\mathbf{d}_s^n$ denote the m-th and n-th descriptors within the flattened maps $\mathbf{D}_g$ and $\mathbf{D}_s$, respectively. The indices $m \in \{1,\dots,H_gW_g\}$ and $n \in \{1,\dots,H_sW_s\}$ represent the 1D spatial positions, and τ is a temperature hyperparameter. Since not all features are matchable, we employ a dual-softmax operator with a learnable dustbin parameter z. The matching probability $\mathbf{M}_{mn}$ is then computed as:

$$\mathbf{M}_{mn} = \frac{\exp(Sim_{mn})}{\sum_{k=1}^{H_sW_s}\exp(Sim_{mk})+\exp(z)} \cdot \frac{\exp(Sim_{mn})}{\sum_{l=1}^{H_gW_g}\exp(Sim_{ln})+\exp(z)}. \tag{2}$$

Inspired by [2], we then compute the final matching confidence weight matrix $\mathbf{W}_{mn} = \mathbf{M}_{mn}\cdot\mathbf{k}_g^m\cdot\mathbf{k}_s^n$, where $\mathbf{k}_g^m$ and $\mathbf{k}_s^n$ denote the keypoints' confidence scores at indices m and n from the flattened maps $\mathbf{K}_g$ and $\mathbf{K}_s$, respectively. This integrates descriptor matching probability with keypoint selection confidence, enabling the matching module to favor correspondences supported by both strong

descriptor similarity and reliable keypoint predictions. Based on $\{\mathbf{W}_{mn}\}$, we select the top-N correspondences with the highest similarity scores across all possible combinations of ground and satellite keypoints. Consequently, the i-th selected correspondence ($i \in \{1, \ldots, N\}$) consists of the satellite 2D keypoint $\mathbf{x}_i$, corresponding ground keypoint $\mathbf{u}_i$ and the matching confidence weight w_i.

3.3 Azimuthal ray generation

For each 2D ground-to-2D satellite keypoint correspondence, the respective azimuthal ray $\mathbf{l}_i \in \mathbb{P}^2$ is constructed by drawing a line on the satellite map crossing the satellite keypoint ($\mathbf{x}_i$) along the 2D direction vector $\mathbf{r}(\phi_i) \in \mathbb{R}^2$, which has a slope of $\tan(\pi/2 - \phi_i)$ with respect to the global x-axis of the North-up satellite map. This yields a 2D point-to-line geometric constraint between the aerial keypoint and the azimuthal ray, expressed as

$$\mathbf{l}_i = \mathbf{x}_i + \lambda \mathbf{r}(\phi_i), \quad \mathbf{l}_i^\top \mathbf{x}_i = 0, \tag{3}$$

where $\lambda \in \mathbb{R}$ denotes a scaling parameter that spans the entire infinite line. As the camera location $\mathbf{p}$ ideally exists at the intersection of all azimuthal rays $\{\mathbf{l}_i\}_{i=1}^N$, we can formulate as $\mathbf{p} = \bigcap_{i=1}^N \mathbf{l}_i$.

3.4 Localization with the ARC solver

After obtaining the 2D ground line-to-2D aerial point correspondences and their geometric constraints, ARC-Loc estimates the camera position through a closed-form solution derived in Eq. (5). To this end, we introduce the geometric formulation of our ARC minimal solver, followed by robust estimation. Then, we describe a detailed strategy to handle noisy orientation priors in real-world cases.

ARC solver. As mentioned in Sec. 3.3, the ground camera ideally lies exactly at the intersection of the mapped azimuthal rays. In practice, however, correspondences inevitably contain false-positive matches that prevent the strict ray convergence at single point. To address this, our ARC solver relaxes the strict intersection constraint into a weighted line-to-point distance minimization problem, where the confidence weight w_i implicitly prioritizes reliable correspondences during optimization. Thus, the optimal camera position $\hat{\mathbf{p}}$ is derived as:

$$\hat{\mathbf{p}} = \underset{\mathbf{p}}{\operatorname{argmin}} \sum_{i=1}^{N} w_i \cdot \operatorname{dist}(\mathbf{p}, \mathbf{l}_i)^2, \tag{4}$$

where $\operatorname{dist}(\cdot, \cdot)$ and N denote the perpendicular line-to-point distance and number of correspondences, respectively. To effectively solve the objective function in Eq. (4), we cast the minimization as a weighted linear least-squares problem. We first define the unit normal vector of the i-th azimuthal line as $\mathbf{n}_i = [\sin(\phi_i), \cos(\phi_i)]^\top$. Then, the perpendicular distance from the candidate position $\mathbf{p}$ to the i-th azimuthal ray ($\mathbf{l}_i$) can be calculated by projecting the vector ($\mathbf{p} - \mathbf{x}_i$) onto the line's normal vector $\mathbf{n}_i$, which is expressed as $\mathbf{n}_i^\top(\mathbf{p} - \mathbf{x}_i)$.

By substituting this into Eq. (4), we can derive the optimal position $\hat{\mathbf{p}}$ in a closed form as

$$\hat{\mathbf{p}} = (\mathbf{A}^\top \mathbf{\Omega} \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{\Omega} \mathbf{b}, \tag{5}$$

$\mathbf{A} \in \mathbb{R}^{N \times 2}$ is the design matrix with its i-th row as $\mathbf{n}_i^\top$, $\mathbf{b} \in \mathbb{R}^N$ is the target vector with $b_i = \mathbf{n}_i^\top \mathbf{x}_i$, and $\mathbf{\Omega} = \mathrm{diag}(w_1, \ldots, w_N)$ is the diagonal weight matrix.

Note, Eq. (5) implies two key aspects. First, $\hat{\mathbf{p}}$ can be estimated using two minimal correspondences. Second, since $\hat{\mathbf{p}}$ is a closed-form solution of $\{w_i\}$, this can be efficiently used for end-to-end training of our pipeline via pose supervision. The full derivation of Eq. (5) is provided in the supplement [11].

Robust estimation. During the inference, as cross-view matching inherently produces a large number of false-positive matches, we equip the ARC-solver with the RANSAC [7] for robust position estimation. To select the best position hypothesis $\mathbf{p}_{\mathrm{hyp}}$, we use a line-to-point distance-based score function defined as:

$$\mathcal{S}(\mathbf{p}_{\mathrm{hyp}}) \;=\; \sum_{i=1}^{N} \frac{w_i}{1 + (\mathrm{dist}(\mathbf{p}_{\mathrm{hyp}}, \mathbf{l}_i)/\sigma)^2}, \tag{6}$$

where σ is a scaling parameter and $\mathbf{l}_i$ denotes the i-th azimuthal line. During inference, as the ARC solver requires a minimal two line-point correspondences to generate $\mathbf{p}_{\mathrm{hyp}}$, we sample multiple hypotheses and select the one maximizing $\mathcal{S}(\mathbf{p}_{\mathrm{hyp}})$ as the initial position. The final position is then obtained via Eq. (5) using all inlier correspondences with the initial position.

Extension for handling noisy orientation. While the global orientation can be obtained from consumer-grade IMUs or orientation prediction networks, these initial estimates inevitably contain some degree of noise [23]. To handle such perturbed conditions, we generate a set of fine-grained heading candidates ($\{\phi_{hyp}\}$) by dividing the search range $[\phi_0 - \Delta,\ \phi_0 + \Delta]$ into equally spaced intervals, where ϕ_0 denotes the noisy orientation prior, and Δ denotes the expected deviation of the orientation noise. For each candidate, we update the sets of azimuthal rays $\{\mathbf{l}_i\}_{i=1}^N$ in Eq. (3) with candidate orientation (ϕ_{hyp}). Then, for each candidate, we estimate the corresponding position ($\mathbf{p}_{\mathrm{hyp}}$) with RANSAC using the quality score in Eq. (6). Consequently, this process yields a set of paired hypotheses ($\phi_{\mathrm{hyp}}, \mathbf{p}_{\mathrm{hyp}}$). We then select the best orientation candidate that maximizes Eq. (6) and use its associated camera position as the final estimation.

Although executing RANSAC for multiple heading candidates may seem to incur high computational cost at first instance, we can efficiently mitigate this overhead at the implementation level by leveraging a parallelized RANSAC framework on modern GPUs such as NVIDIA A100.

3.5 Loss function

To train the ARC-Loc framework in an end-to-end manner, we introduce the Azimuthal Ray Convergence (ARC) loss (Fig. 3) $\mathcal{L}_{\mathrm{ARC}}$, which is a composite of the

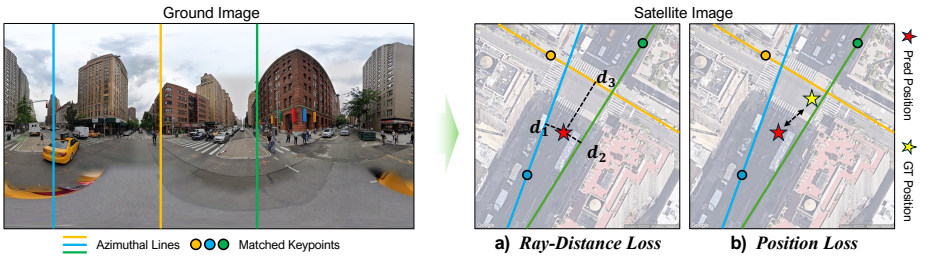

Fig. 3: ARC loss visualization. Our ARC loss aims to supervise the ARC-Loc framework in an end-to-end manner. (a) The ray-distance loss enforces the azimuthal lines to converge at the predicted location without requiring explicit matching labels, while (b) the position loss directly minimizes the discrepancy between the predicted and ground-truth camera positions. By backpropagating these geometric errors, the network implicitly learns feature matching by penalizing geometrically inconsistent outliers.

position loss ($\mathcal{L}_{position}$) and the (azimuthal) ray-distance loss ($\mathcal{L}_{ray-dist}$). Specifically, $\mathcal{L}_{\text{ARC}} = \mathcal{L}_{position} + \alpha\mathcal{L}_{ray-dist}$, where α is a hyperparameter to balance the two terms. $\mathcal{L}_{position}$ is defined as $\|\mathbf{p}_{gt} - \hat{\mathbf{p}}\|_2^2$, where $\mathbf{p}_{gt}$ is the ground-truth position and $\hat{\mathbf{p}}$ is the estimated position, providing direct supervision signal for the camera location. On the other hand, $\mathcal{L}_{ray-dist}$ is defined as $\sum_{i=1}^{N} \bar{w}_i \cdot \text{dist}(\hat{\mathbf{p}}, \mathbf{l}_i)^2$, where $\{\bar{w}_i\}$ represent a set of normalized weights obtained by applying the softmax operation to the matching scores $\{w_i\}$. By leveraging the solver's predicted position $\hat{\mathbf{p}}$ as a geometric anchor, this objective explicitly penalizes rays that fail to converge. Since this loss is weighted by the matching confidence $\bar{w}_i$, the keypoint and descriptor heads (Sec. 3.2) learn to lower the matching scores of correspondences yielding geometrically inconsistent azimuthal rays, acting as an effective geometric regularization technique for cross-view localization.

4 Experiments

4.1 Datasets and evaluation metrics

VIGOR dataset. VIGOR [34] is a benchmark for fine-grained cross-view localization, comprising 360° ground panoramas and 70 m × 70 m geo-referenced satellite images from four cities (Chicago, NewYork, SanFrancisco, Seattle). It provides Same-Area (unseen locations) and Cross-Area (unseen cities) settings for evaluation. To assess orientation robustness, we use both orientation-aligned panoramas and those with noisy orientations applied via horizontal cyclic shifts.

KITTI dataset. KITTI [8,18,21] is a widely used benchmark featuring perspective images from front-facing cameras paired with 100 m × 100 m geo-referenced satellite maps. An orientation prior is typically provided with an added noise level of approximately ±10° to simulate the inaccuracies of consumer-grade IMU/compass. The dataset is partitioned into Training, Test1, and Test2 sets. While the Training and Test1(Same-Area) sets consist of different measurements

Table 1: Localization results on the VIGOR dataset under the known orientation setting. The best and second-best results within each category (BEV and Non-BEV) are highlighted in **bold** and underlined, respectively.

Type	Models	Same-Area		Cross-Area	
		↓ Localization (m)		↓ Localization (m)	
		Mean	Median	Mean	Median
BEV	GGCVT	4.12	1.34	5.16	1.40
	DenseFlow	3.03	**0.97**	5.01	2.42
	G2SWeakly	4.19	1.68	4.70	1.68
	HC-Net	<u>2.65</u>	1.17	3.35	1.59
	FG^2	**1.95**	<u>1.08</u>	**2.41**	<u>1.37</u>
	BevSplat	2.87	1.58	<u>2.84</u>	**1.36**
Non-BEV	SliceMatch	5.18	2.58	5.53	2.55
	CCVPE	3.60	<u>1.36</u>	4.97	<u>1.68</u>
	Loc^2	<u>3.06</u>	1.59	<u>3.43</u>	1.90
	ARC-Loc (ours)	**2.52**	**1.20**	**3.11**	**1.59**

taken from the same region in the training data sequence, the Test2(Cross-Area) set contains data from unseen regions.

Metrics. We evaluate ARC-Loc using mean and median position and orientation errors on both datasets. For KITTI, we additionally report recall at position ($R@1m, R@5m$) and orientation ($R@1°, R@5°$) thresholds, and decompose the position error into longitudinal and lateral components. We further assess localization accuracy given known orientations as well as under noisy prior conditions designed to reflect the practical error ranges of consumer-grade IMUs. Finally, we report inference latency and memory usage to evaluate practical deployability.

Implementation details. We employ pretrained RADIOv3-L [16] as our feature backbone, keeping weights frozen to preserve generalized representations. Input images are resized to 720×1440 for ground panoramas and 656×656 for satellite images. Both descriptor and keypoint heads project features into a 128-D space. During matching, we compute cosine similarity with a temperature of $\tau = 0.1$ and select the top $N = 512$ correspondence pairs to generate azimuthal rays. The model is trained using the Adam optimizer with the initial learning rate of 10^{-4} and a Cosine Annealing scheduler. We set the ARC loss coefficient α to 0.5 and train with a batch size of 64 on a single NVIDIA A100 GPU. For the KITTI dataset, we adopt the orientation estimator from [19].

4.2 Localization results

To demonstrate the effectiveness of our proposed ARC-Loc, we compare ARC-Loc against the state-of-the-art cross-view localization methods, including BEV-based approaches [19, 20, 22, 25, 27, 28] and Non-BEV approaches [12, 29, 31].

VIGOR. In Tab. 1, we first evaluate ARC-Loc under the known orientation setting. Notably, our method achieves the lowest position error among all Non-BEV

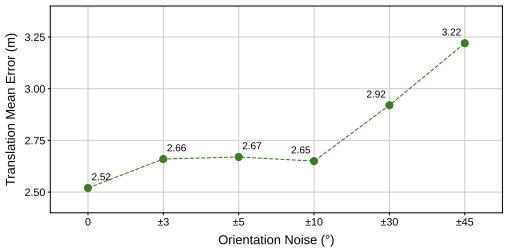

Fig. 4: Impact of orientation noise. Mean position error under bounded heading uncertainty ($\pm\Delta^\circ$), accounting for IMU errors ($\pm 5.9^\circ$) [23] and environmental disturbances.

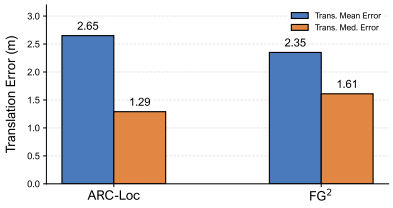

Fig. 5: Position mean and median errors of ARC-Loc versus FG^2 given $\pm 10^\circ$ orientation noise.

methods. Specifically, it outperforms Loc^2 without relying on any external depth modules, reducing the mean localization error by 17.6% on the Same-Area and by 9.3% on the Cross-Area. Compared to BEV-based methods, ARC-Loc yields a slightly higher error than the state-of-the-art FG^2 [28], while still exhibiting lower mean position errors than several BEV-based approaches, demonstrating its highly competitive localization accuracy. In Overall, ARC-Loc achieves the second-best overall mean error on the Same-Area setting, closely following FG^2. We visually confirm this mechanism in Fig. 6, under the known orientation setting, the extracted azimuthal rays converge around the ground-truth location, highlighting the interpretability of our framework without explicit 3D lifting.

While the previous results validate the effectiveness of our position-focused design under known orientation, real-world deployment often involves noisy sensor data. To evaluate our framework's robustness in such practical scenarios, we further evaluate ARC-Loc under bounded heading uncertainties. Fig. 4 demonstrates our model's stability, showing that the mean position error remains within 2.65 m for noise levels up to $\pm 10^\circ$, before gradually increasing at larger noise bounds (e.g., $\pm 45^\circ$). Notably, despite having a higher mean position error than FG^2 under the practical $\pm 10^\circ$ noisy orientation setting (Fig. 5), ARC-Loc achieves a lower median position error of 1.29 m. This confirms that our azimuthal ray based approach is effective even under orientation noise.

KITTI. We evaluate ARC-Loc against both BEV-based and Non-BEV methods. As shown in Tab. 2, ARC-Loc demonstrates competitive position accuracy on the Same-Area setting, performing on par with leading BEV-based approaches like FG^2 and outperforming existing Non-BEV baselines. On the Cross-Area setting, our method yields a higher position error than Loc^2, but it maintains stable performance. This result can be attributed to our line-based geometric formulation, which relies on precise azimuthal cues that are inherently more challenging to extract under the restricted FoV and unseen layouts of the Cross-Area setting. Nevertheless, these findings suggest that ARC-Loc effectively balances computational efficiency with localization accuracy, positioning it as a feasible alternative to more computationally intensive models.

Table 2: Quantitative comparison on the KITTI dataset under orientation noise ($\pm 10^{\circ}$). The best and second-best results within each category (BEV and Non-BEV) are highlighted in **bold** and underlined, respectively. * denotes the use of the orientation estimator from G2SWeakly [19].

Type	Models	Loc. (m) ↓		Lateral (%)↑		Long. (%)↑		Ori. (°) ↓		Ori. ↑	
		Mean	Median	R@1m	R@5m	R@1m	R@5m	Mean	Median	R@1°	R@5°
Same-Area											
BEV	GGCVT	-	-	76.44	98.89	23.54	62.18	-	-	99.10	**100.0**
	DenseFlow	1.48	**0.47**	95.47	**99.79**	87.89	94.78	0.49	0.30	89.40	99.31
	HC-Net	0.80	0.50	**99.01**	99.73	92.20	**99.25**	0.45	0.33	91.35	99.84
	FG^2	**0.75**	0.51	95.81	99.66	**92.50**	99.05	0.93	0.66	67.27	98.91
	BevSplat*	2.87	2.06	60.28	94.24	35.62	76.57	**0.33**	**0.28**	**99.99**	**100.0**
Non-BEV	CCVPE	1.22	0.62	97.35	99.71	77.13	97.16	0.67	0.54	77.39	99.95
	Loc^2	1.13	0.77	93.59	**99.97**	71.51	98.17	1.97	1.43	36.68	92.84
	ARC-Loc* (ours)	**0.79**	**0.50**	**97.72**	99.63	**87.01**	**98.60**	**0.21**	**0.18**	**99.87**	**100.0**
Cross-Area											
BEV	GGCVT	-	-	57.72	91.16	14.15	45.00	-	-	98.98	**100.0**
	DenseFlow	7.97	3.52	54.19	91.74	23.10	61.75	2.17	1.21	43.44	89.31
	HC-Net	8.47	4.57	**75.00**	**97.76**	**58.93**	**76.46**	3.22	1.63	33.58	83.78
	FG^2	7.31	4.15	37.89	85.65	21.98	60.77	3.62	2.37	23.03	77.84
	BevSplat*	**6.20**	**2.51**	60.01	95.17	27.41	60.45	**0.33**	**0.28**	**99.99**	**100.0**
Non-BEV	CCVPE	9.16	3.33	44.06	90.23	23.08	64.31	1.55	0.84	57.72	96.19
	Loc^2	**5.60**	**3.01**	**45.29**	**92.43**	**27.01**	**68.26**	3.32	2.12	26.03	80.68
	ARC-Loc* (ours)	7.45	4.32	38.45	87.13	20.26	59.85	**0.20**	**0.18**	**99.93**	**100.0**

Inference and memory efficiency. To evaluate practicality on resource-constrained platforms, we compare inference latency and memory consumption on VIGOR across inference configurations (Tab. 3). Here, matching-stage latency includes BEV feature construction or depth inference when required. Since ARC-Loc performs direct matching in the native 2D image domain without BEV transformation or depth foundation modules, it achieves substantially lower matching latency while maintaining a compact memory footprint compared to depth-dependent methods such as Loc^2 and BevSplat. These results show that ARC-Loc improves efficiency by removing the overhead of BEV construction and depth-based 3D lifting, while maintaining competitive localization accuracy.

Table 3: Inference latency and memory comparison with ViT-based models. Matching Latency and Pose Est. Latency denote the feature matching and solver/RANSAC stages, respectively. Original denotes each method's official sequential RANSAC implementation, while Parallel (ours) denotes our GPU-parallel RANSAC implementation.

Method	**BEV**	**Depth Module**	**RANSAC**	**Matching ↓ Latency (ms)**	**Pose Est. ↓ Latency (ms)**	**Memory (MB) ↓**
FG^2	✓	✗	✗	187.61	2.03	885
			Original	185.94	3630	910
			Parallel (ours)	187.34	39.67	1281
BevSplat	✓	✓	–	139.90 (end-to-end)		2577
Loc^2	✗	✓	✗	137.98	1.44	2131
			Original	136.13	3984	2168
			Parallel (ours)	137.52	28.32	2810
ARC-Loc	✗	✗	✗	58.69	1.70	906
			Parallel (ours)	59.67	20.03	1134

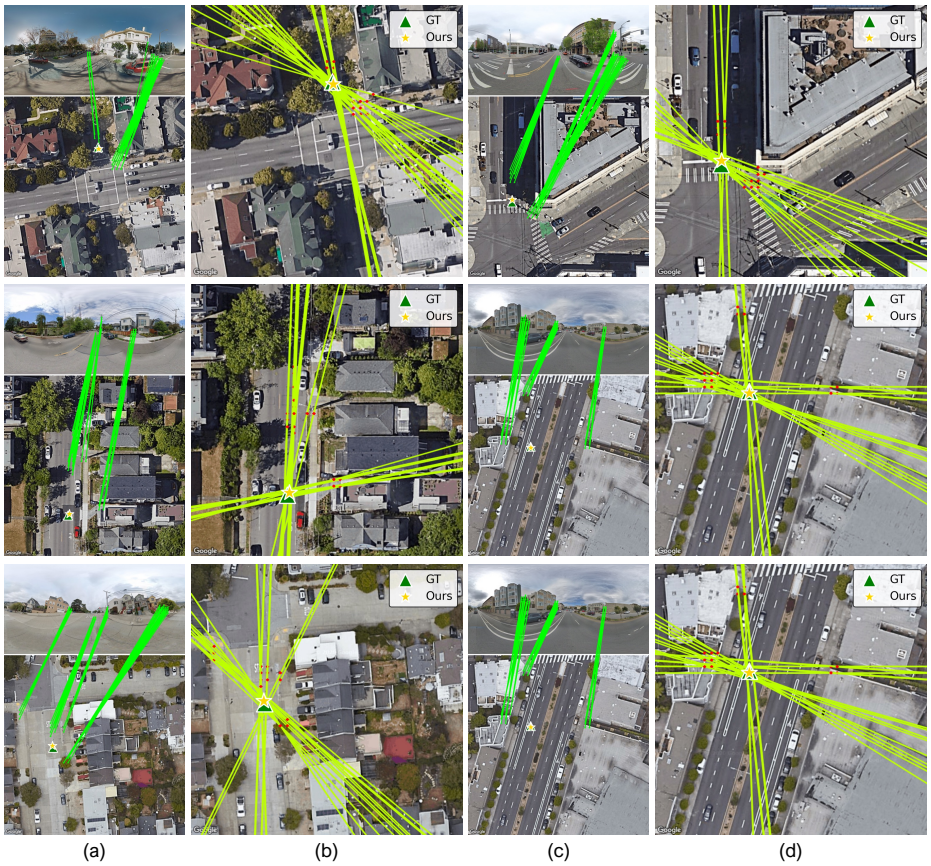

Fig. 6: Visualization of feature matching and azimuthal rays. (a), (c) denote cross-view feature matching, and (b), (d) denote azimuthal rays for corresponding matches.

4.3 Ablation study

We conduct ablation studies on (i) the feature extraction backbone and (ii) hyperparameter settings. All experiments are conducted using the VIGOR dataset under the Same-Area.

Backbone. We evaluate ARC-Loc's sensitivity to the feature extractor by replacing our default RADIOv3 [16] with DINOv2 [14], the standard backbone for recent baselines like FG^2, Loc^2, and BevSplat. As shown in Tab. 4, RADIOv3 yields lower mean and median errors and faster matching-stage latency, consistent with its stable performance across context module configurations (Sec. 3.3), likely due to its distilled training and high-resolution fine-grained cues. ARC-Loc still remains robust and computationally efficient with DINOv2, demonstrating that our framework is not tied to a specific foundation backbone. This suggests that the proposed azimuthal ray-based formulation remains effective with different feature extractors, while the choice of backbone further affects localization accuracy and efficiency.

Table 4: Backbone ablation on VIGOR Same-Area. Match Lat. denotes matching-stage latency (ms), and Mem. denotes memory usage (MB), both measured under the parallel RANSAC setting.

Backbone	Loc. (m) ↓		Match Lat. ↓	Mem. ↓
	Mean	Median		
RADIOv3	**2.52**	**1.20**	**59.67**	1134
DINOv2	2.90	1.40	94.53	**1119**

Table 5: Ablation study on hyperparameters. The default is $\alpha = 0.5$ and $N = 512$.

Setting	Loc. (m) ↓	
	Mean	Median
$\alpha = 0$	2.76	1.23
$\alpha = 0.1$	2.76	1.24
$\alpha = 1.0$	3.98	2.50
Default ($\alpha = 0.5, N = 512$)	**2.52**	**1.20**
$N = 256$	4.57	3.01
$N = 1024$	2.67	1.21

Loss functions and hyperparameters. We conduct an ablation study on the ray-distance loss weight α and the number of selected matches N to optimize performance under severe cross-view outliers (Tab. 5). While $\mathcal{L}_{\text{position}}$ alone yields a 2.76 m mean error, the integration of $\mathcal{L}_{ray-dist}$ with our default setting ($\alpha = 0.5, N = 512$) is observed to improve accuracy to 2.52 m. The weight α appears to balance the primary objective and geometric regularization; a minimal weight ($\alpha = 0.1$) provides negligible gains, while an excessive weight ($\alpha = 1.0$) may over-regularize the network, distracting it from the primary pose estimation objective and degrading performance to 3.98 m.

Similarly, the match count N represents a trade-off between structural diversity and outlier noise. A limited count ($N = 256$) appears to provide insufficient context for effective learning (4.57 m), whereas an overly large count ($N = 1024$) may introduce low-confidence matches that corrupt the training signal (2.67 m). Consequently, $N = 512$ is suggested as the optimal empirical balance for achieving robust localization.

5 Conclusion

In this paper, we presented ARC-Loc, a novel cross-view localization framework that bypasses the conventional dependencies on Bird's-Eye-View transformations and external depth foundation models. Inspired by the traditional human navigation technique of *resection*, we reformulated ground-to-aerial correspondences as 2D line-to-point constraints and introduced the Azimuthal Ray Convergence (ARC) solver, a closed-form 2-point minimal solver that estimates the camera location directly within the native 2D image domain. Additionally, we proposed the ARC Loss to optimize the matching network by enforcing ray convergence without requiring dense correspondence labels or depth priors.

Through evaluations on the VIGOR and KITTI datasets, we demonstrated that ARC-Loc achieves competitive localization accuracy compared to state-of-the-art methods. By operating entirely in 2D, we showed our approach benefits from reduced inference latency and memory consumption compared to recent frameworks relying on 3D lifting or BEV projections, highlighting its practical viability for resource-constrained autonomous platforms.

Limitations. While ARC-Loc offers an efficient alternative for cross-view localization, it presents two primary limitations. First, although the framework remains robust within practical orientation noise bounds (*e.g.*, up to $\pm 10^{\circ}$), position accuracy gradually degrades under larger heading uncertainties. Second, the stability of the ARC solver relies on the directional diversity of the azimuthal rays. In scenarios with a restricted field-of-view, such as the front-facing cameras in *KITTI*, matched keypoints often concentrate in a narrow region, lacking angular diversity and leading to poorly constrained ray intersections (geometric degeneracy). Addressing these remains an important direction for future research.

Acknowledgements. This work was supported by National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIT) (No. RS-2026-25486241, 50%), by Ministry of Trade, Industry and Energy (MOTIE) and the Korea Institute for Advancement of Technology (KIAT) under the Corporate Demand-Driven Challenge and Innovative R&D Program for Next-Generation Researchers (Grant No. RS-2026-25539646, 25%), and by Institute of Information & Communications Technology Planning & Evaluation (IITP) under the Artificial Intelligence Semiconductor Support Program to Nurture the Best Talents (IITP-(2026)-RS-2023-00253914, 25%) grant funded by the Korea government (MSIT).

References

1. Army, D.: Map Reading and Land Navigation: FM 3-25.26. Field manual, CreateSpace Independent Publishing Platform (2011), `https://books.google.co.kr/books?id=XaKcCgAAQBAJ`
2. Barroso-Laguna, A., Munukutla, S., Prisacariu, V.A., Brachmann, E.: Matching 2d images in 3d: Metric relative pose from metric correspondences. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4852–4863 (2024)
3. Ben-Moshe, B., Elkin, E., Levi, H., Weissman, A.: Improving accuracy of gnss devices in urban canyons. In: CCCG. pp. 511–515 (2011)
4. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)
5. CHROBOTICS: UM7 Orientation Sensor Datasheet (Revision 1.3). CHROBOTICS (nd), `https://www.pololu.com/file/0J773/UM7_Datasheet_1-3.pdf`, manufacturer datasheet with typical yaw accuracy specifications
6. Fervers, F., Bullinger, S., Bodensteiner, C., Arens, M., Stiefelhagen, R.: Uncertainty-aware vision-based metric cross-view geolocalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21621–21631 (2023)
7. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)
8. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. International Journal of Robotics Research (IJRR) (2013)

9. Hu, S., Feng, M., Nguyen, R.M., Lee, G.H.: Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7258–7267 (2018)
10. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023)
11. Kim, H., Kim, M., Moon, H., Hong, J.H.: Supplementary material for ARC-Loc: Leveraging Azimuthal Ray Convergence as a Geometric Cue for Direct Cross-View Localization (2026)
12. Lentsch, T., Xia, Z., Caesar, H., Kooij, J.F.: Slicematch: Geometry-guided aggregation for cross-view pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17225–17234 (2023)
13. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Yu, Q., Dai, J.: Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
14. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. Transactions on Machine Learning Research Journal pp. 1–31 (2024)
15. Piccinelli, L., Sakaridis, C., Segu, M., Yang, Y.H., Li, S., Abbeloos, W., Van Gool, L.: Unik3d: Universal camera monocular 3d estimation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 1028–1039 (2025)
16. Ranzinger, M., Heinrich, G., Kautz, J., Molchanov, P.: Am-radio: Agglomerative vision foundation model reduce all domains into one. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12490–12500 (2024)
17. Schönemann, P.H.: A generalized solution of the orthogonal procrustes problem. Psychometrika **31**(1), 1–10 (1966)
18. Shi, Y., Li, H.: Beyond cross-view image retrieval: Highly accurate vehicle localization using satellite image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2022)
19. Shi, Y., Li, H., Perincherry, A., Vora, A.: Weakly-supervised camera localization by ground-to-satellite image registration. In: European Conference on Computer Vision. pp. 39–57. Springer (2024)
20. Shi, Y., Wu, F., Perincherry, A., Vora, A., Li, H.: Boosting 3-dof ground-to-satellite camera localization accuracy via geometry-guided cross-view transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 21516–21526 (2023)
21. Shi, Y., Yu, X., Wang, S., Li, H.: Cvlnet: Cross-view semantic correspondence learning for video-based camera localization. In: Asian Conference on Computer Vision. pp. 123–141. Springer (2022)
22. Song, Z., Lu, J., Shi, Y., et al.: Learning dense flow field for highly-accurate cross-view camera localization. Advances in Neural Information Processing Systems **36**, 70612–70625 (2023)
23. STMicroelectronics: LSM6DSV: Inertial Measurement Unit Datasheet. STMicroelectronics (nd), `https://www.st.com/resource/en/datasheet/lsm6dsv.pdf`, manufacturer datasheet for IMU specifications
24. Vo, N.N., Hays, J.: Localizing and orienting street views using overhead imagery. In: European conference on computer vision. pp. 494–509. Springer (2016)
25. Wang, Q., Wu, S., Shi, Y.: Bevsplat: Resolving height ambiguity via feature-based gaussian primitives for weakly-supervised cross-view localization. In: Advances in

Neural Information Processing Systems (NeurIPS) (2026), https://arxiv.org/abs/2502.09080
26. Wang, S., Nguyen, C., Liu, J., Zhang, Y., Muthu, S., Maken, F.A., Zhang, K., Li, H.: View from above: Orthogonal-view aware cross-view localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14843–14852 (2024)
27. Wang, X., Xu, R., Cui, Z., Wan, Z., Zhang, Y.: Fine-grained cross-view geo-localization using a correlation-aware homography estimator. Advances in Neural Information Processing Systems **36**, 5301–5319 (2023)
28. Xia, Z., Alahi, A.: Fg^2: Fine-grained cross-view localization by fine-grained feature matching. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 6362–6372 (2025)
29. Xia, Z., Booij, O., Kooij, J.F.: Convolutional cross-view pose estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence **46**(5), 3813–3831 (2023)
30. Xia, Z., Booij, O., Manfredi, M., Kooij, J.F.: Visual cross-view metric localization with dense uncertainty estimates. In: European Conference on Computer Vision. pp. 90–106. Springer (2022)
31. Xia, Z., Xu, C., Alahi, A.: Loc^2: Interpretable cross-view localization via depth-lifted local feature matching. arXiv preprint arXiv:2509.09792 (2025)
32. Yuan, D., Maire, F., Dayoub, F.: Cross-attention between satellite and ground views for enhanced fine-grained robot geo-localization. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1249–1256 (2024)
33. Zang, A., Xu, R., Trajcevski, G., Zhou, F.: Data issues in high-definition maps furniture–a survey. ACM Transactions on Spatial Algorithms and Systems **10**(1), 1–37 (2024)
34. Zhu, S., Yang, T., Chen, C.: Vigor: Cross-view image geo-localization beyond one-to-one retrieval. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3640–3649 (2021)