BEV-Patch-PF: Particle Filtering with BEV-Aerial Feature Matching for Off-Road Geo-Localization

Dongmyeong Lee*, Jesse Quattrociocchi†, Christian Ellis†, Rwik Rana*,
Amanda Adkins*, Adam Uccello†, Garrett Warnell†, Joydeep Biswas*

*The University of Texas at Austin †DEVCOM Army Research Laboratory

***Abstract*—Accurate localization of ground robots using aerial imagery is essential for off-road navigation and planning, especially in GPS-denied environments. However, this task remains challenging due to large viewpoint differences, scarce distinctive features, and high environmental variability. Most existing approaches typically localize each frame independently, either by retrieving global descriptors or by aligning ground and aerial features in a shared spatial representation, making them susceptible to ambiguity and multi-modal pose estimates. While sequential localization can reduce such uncertainty, existing per-frame methods incur trade-offs between accuracy, memory, and computational cost, limiting their effectiveness in a sequential setting.**

We propose BEV-Patch-PF, a GPS-free sequential localization system that integrates a particle filter with a learned bird's-eye-view (BEV) observation model. For each particle pose hypothesis, a single aerial feature patch is cropped and its likelihood is computed by comparing it against the BEV feature derived from the on-board view. Ground features are extracted using a visual foundation model, and fused with aerial features via cross-attention to emphasize salient off-road regions Experiments on real-world off-road routes from the TartanDrive 2.0 dataset demonstrate that BEV-Patch-PF outperforms stereo visual odometry and a retrieval-based baseline in trajectory accuracy across both seen and unseen environments, highlighting its robustness and generalization.

I. Introduction

Aerial imagery offers global context essential for safe off-road robot navigation, enabling path planning around natural hazards such as cliffs, rivers, and dense vegetation. However, leveraging such imagery requires accurate geo-referenced localization within an aerial map, which remains challenging in off-road environments where GPS is often unreliable due to occlusions or interference. Vision-based localization systems such as Visual odometry (VO) can provide short-term pose estimates, but accumulate drift without access to global position fixes, leading to large localization errors that compromise downstream planning and decision-making.

Cross-view geo-localization addresses the lack of global position fixes by estimating a robot's pose through matching ground-level images with geo-referenced aerial imagery. However, this task is inherently difficult due to the large viewpoint difference between ground and aerial images. This problem is especially challenging in unstructured off-road environments, where the absence of structural landmarks—and the presence of terrain irregularities, dense vegetation, and seasonal appearance changes—exacerbates the visual mismatch and removes many of the cues that conventional methods rely on [2, 24].

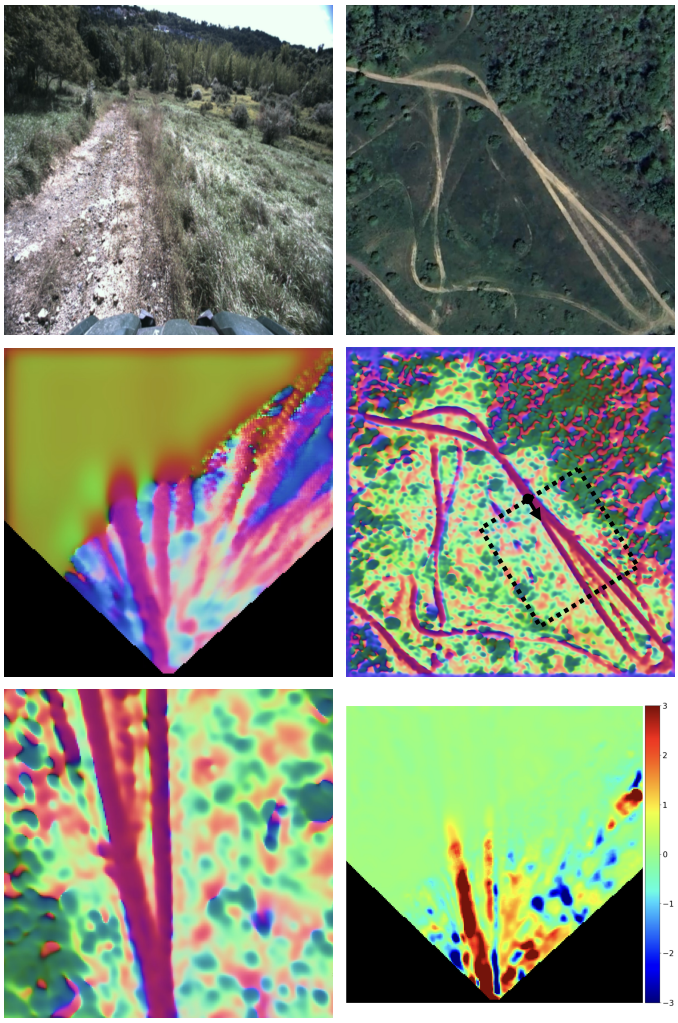


Fig. 1: Visualization of BEV-Patch-PF inputs and outputs. **Top**: On-board camera image and local satellite image. **Middle**: BEV ground features and aerial features; dotted box and arrow indicate the hypothesized patch and pose, respectively. **Bottom**: Extracted aerial patch and its dot-product heatmap.

Recent deep learning approaches tackle this problem frame-by-frame and fall into two categories: (1) retrieval-based methods [6, 18, 33, 28, 29, 34, 9], which learn global descriptors for ground images and aerial patches; and (2) spatial

feature-alignment methods [19, 3, 20, 23, 21], which learn features from ground and aerial imagery within a shared spatial representation, inferring the pose best aligns those features. Per-frame localization, however, considers only a single observation at a time, making it vulnerable to perceptual ambiguity and multi-modal solutions. In off-road settings, this can lead to catastrophic pose jumps—caused by visually similar map regions, sensor occlusions, or accumulated noise. Sequential localization mitigates these issues by jointly reasoning over temporal pose sequences, reducing ambiguity through temporal consistency.

While sequential inference can reduce pose ambiguity, it requires observation models that yield smooth, discriminative likelihoods over continuous pose hypotheses. Existing cross-view methods [29, 34, 19, 3] were not designed with this requirement. Retrieval-based approaches assign coarse similarity scores over a discretized set of aerial patches, making them insensitive to fine-grained pose changes and unsuitable for continuous probabilistic filtering. In contrast, spatial feature-alignment methods offer improved granularity but they either: (i) require dense correlation over discretized pose grids—leading to high computational cost or (ii) or optimize directly for a single best pose, which is difficult to integrate as a likelihood over pose hypotheses.

To address these limitations, we introduce a sequential localization system that integrates a particle filter with an observation model capable of evaluating likelihoods over continuous pose hypotheses. On-board images are projected into to bird's-eye-view (BEV) feature maps using a visual-foundation backbone [15] and fused with aerial features via cross-attention. For each particle, an oriented aerial feature patch is extracted and compared against the BEV feature map using a point-wise dot product. Because aerial patches can be sampled at arbitrary continuous poses, the observation likelihood is directly computed at each particle's continuous hypothesis, making it a natural fit for particle filtering. The model is trained specifically for unstructured off-road terrain, without relying on semantic landmarks.

We evaluated our approach on real-world off-road trajectories and compare against two baselines: stereo visual odometry [14], and a retrieval-based pose-graph-optimization method [9]. Across both seen and unseen routes from the TartanDrive 2.0 [22] dataset, our method consistently achieves lower trajectory error and greater robustness. These results demonstrate the benefits of continuous-pose likelihood modeling while confirming the method's ability to generalize to previously unobserved environments without requiring GPS.

II. Related Works

Visual geo-localization estimates a robot's 3-DoF pose within a geo-referenced map from a ground-level image. Most approaches formulate the task as visual place recognition (VPR), retrieving the most similar geo-tagged ground image from a pre-collected database and transferring its pose [8, 13]. Although effective in densely imaged urban areas, VPR scales poorly and is impractical for off-road missions, because no prior data collection can be performed.

Cross-View Geo-Localization tackles the same objective without ground database by matching each ground image directly to overhead imagery – satellite photos or semantically labeled planimetric maps. Most methods learn cross-view descriptors [6, 18, 33, 28, 29, 34, 9] through contrastive learning, pulling a ground-image embedding toward that of the aerial patch at its ground-truth pose. Yet their accuracy is limited by the density of sampled aerial patches and by the absence of explicit orientation modeling. Later work [30, 10] encodes multiple rotations per grid cell to infer heading, but the estimates remain coarse.

Spatial-feature-alignment methods then emerged: (1) dense cross-correlation in BEV space [19, 3], (2) continuous-pose optimization [20, 23, 21], and (3) view-synthesis matching [7, 25]. Dense correlation offers the best precision, yet evaluating K rotated kernels at every location of an $H \times W$ feature grid costs $\mathcal{O}(KH^2W^2)$. The optimization and synthesis variants avoid that sweep but are vulnerable to local minima and seasonal appearance drift. Motivated by the dense correlation but seeking lower computational cost, we evaluate the likelihood only at a set of N pose particles, extracting a single aerial patch per hypothesis. This reduces complexity to $\mathcal{O}(NHW)$, eliminates exhaustive sweeps, and treats orientation as a continuous variable. These patch-wise likelihoods serve as the observation model for the particle filter used for sequential localization, which we discuss next.

Sequential Cross-View Localization mitigates the ambiguity and multi-modal solutions of single-frame localization by propagating joint pose probabilities. Particle filter with retrieval-based observation models [28, 5, 32] compute the likelihood by comparing descriptors: the descriptor of the closest grid cell in position and orientation is treated as the expected observation and compared to that of the ground image. However, this map-grid encoding inherits the resolution limits of retrieval-based localization. Sarlin et al. [19] warp dense probability maps temporally to compute joint probability, but require ground-truth odometry. Klammer and Kaess [9] embed the per-frame localization in a pose graph, but need approximate GPS to filter outliers before adding registration factors. The recent end-to-end particle smoother [31] correlates BEV feature against aerial feature map, but is confined to urban scenes and planimetric map.

Off-Road Cross-View Localization remains underexplored. Nearly all existing methods and datasets [28, 5, 32, 31, 19, 33, 1, 11] focus on urban scenes and rely on semantically annotated planimetric maps. In unstructured terrain, such semantic overlays are unavailable, distinctive man-made cues are scarce, and on-board images are often texture-poor, making urban-centric assumptions untenable. To our knowledge, only BEVLoc [9] and BEVRender [7] conduct off-road experiments, both on TartanDrive2.0 [22] using satellite photos.

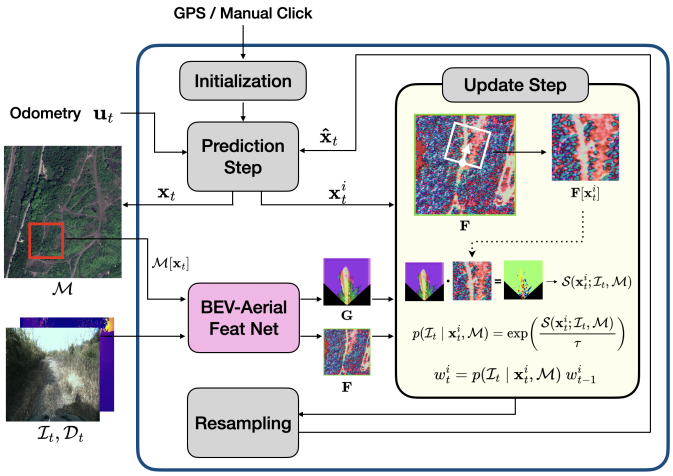


Fig. 2: Overall pipeline of the BEV-Patch-PF.

III. Particle Filtering with BEV-Aerial Feature

We propose BEV-Patch-PF, a sequential localization framework that combines a particle filter with an observation model based on learned bird's-eye-view (BEV) and aerial feature matching. Particles are propagated using egomotion estimates from stereo visual odometry, then reweighted by evaluating how well each particle's predicted aerial-view appearance aligns with the BEV feature map extracted from the current RGB-D observation. Specifically, for each particle, we extract an oriented aerial feature patch and compute a similarity score with the on-board BEV representation, which defines the observation likelihood. This filtering loop allows the system to integrate information over time, correct for accumulated drift, and maintain robust localization even in ambiguous or perceptually aliased environments. The remainder of this section formalizes the pose estimation problem, describes the particle filtering procedure, and details the BEV-aerial feature network used to compute observation likelihoods.

A. Problem Formulation

We track the ground vehicle's planar 3-DoF pose $\mathbf{x}_t = (x_t, y_t, \theta_t) \in \mathrm{SE}(2)$, where (x_t, y_t) are east- and north-directed UTM coordinates (meters) and $\theta_t \in (-\pi, \pi]$ is the heading, measured counter-clockwise from the east axis of a north-up satellite map.

The filter receives: (1) Ego-centric RGB image $\mathcal{I}_t$ from a forward-facing camera with known pinhole camera calibration. (2) Depth image $\mathcal{D}_t$ obtained from stereo and aligned to the RGB frame. (3) Geo-referenced satellite image $\mathcal{M}$ covering the full operating area. (4) Odometry increment $\mathbf{u}_t \in \mathrm{SE}(2)$ between times $t-1$ and t, obtained from stereo visual odometry (interchangeable with wheel, LiDAR, or IMU odometry).

B. Particle Filter Localization

The overall BEV-Patch-PF pipeline is illustrated in Fig. 2.

Initialization. Particles are initialized from approximate GPS coordinates or a user-selected location, then perturbed with Gaussian noise. It is worth noting that GPS is required only for this first step.

Prediction step. Each particle's pose $\mathbf{x}_t^i$ at time t is obtained from its predecessor $\mathbf{x}_{t-1}^i$ by propagating the motion $\mathbf{u}_t$ and adding Gaussian noise to account for odometry error:

$$\mathbf{x}_t = \mathbf{x}_{t-1} \cdot \mathbf{u}_t \cdot \mathbf{w}_\epsilon, \quad \mathbf{w}_\epsilon = \mathrm{Exp}(\boldsymbol{\delta}). \tag{1}$$

Here $\mathbf{w}_\epsilon$ is the prediction noise, $\mathrm{Exp}(\cdot)$ maps $\mathfrak{se}(2)$ to $\mathrm{SE}(2)$, and $\boldsymbol{\delta} \in \mathbb{R}^3$ is a zero-mean Gaussian vector with covariance $\mathrm{diag}(\sigma_t^2, \sigma_t^2, \sigma_\theta^2)$. The parameters σ_t and σ_θ are the standard deviations of the translational and rotational noise, respectively.

Update step. The ego-centric image $\mathcal{I}_t$ updates the particle weights w_t^i via the measurement likelihood $p(\mathcal{I}_t \mid \mathbf{x}_t^i, \mathcal{M})$. A local satellite image $\mathcal{M}[\mathbf{x}_t]$ is cropped from the full aerial map $\mathcal{M}$, centered on pose $\mathbf{x}_t$. The BEV-aerial feature network (described in III-C) produces a BEV feature map $\mathbf{G} \in \mathbb{R}^{H_g \times W_g \times D}$ and a aerial feature map $\mathbf{F} \in \mathbb{R}^{H_a \times W_a \times D}$. We extract patch $\mathbf{F}[\mathbf{x}_t^i] \in \mathbb{R}^{H_g \times W_g \times D}$ oriented and around each particle pose $\mathbf{x}_t^i$ from the $\mathbf{F}$ with the same size of BEV feature. Then compute the likelihood with the point-wise dot product as follow:

$$p(\mathcal{I}_t \mid \mathbf{x}_t^i, \mathcal{M}) = \exp\big(\mathcal{S}(\mathbf{x}_t^i; \mathcal{I}_t, \mathcal{M})/\tau_s\big) \tag{2}$$

$$\mathcal{S}(x_t^i; \mathcal{I}_t, \mathcal{M}) = \frac{1}{H_g W_g} \sum_{v=1}^{H_g} \sum_{u=1}^{W_g} \mathbf{G}_{uv} \cdot \mathbf{F}[\mathbf{x}_t^i]_{uv} \tag{3}$$

When a particle's pose hypothesis is accurate, its oriented aerial patch aligns tightly with the BEV features, producing a high correlation and, after the exponential scaling, a high likelihood. Weights are updated by $w_t^i = p(\mathcal{I}_t \mid \mathbf{x}_t^i, \mathcal{M})\, w_{t-1}^i$ and then normalized to sum up to 1.

Resampling step. Low-variance resampling is triggered only when the effective sample size falls below a preset threshold, keeping the particles most likely to match the true pose and discarding the less plausible ones.

C. Bird's-Eye View & Aerial Feature Network

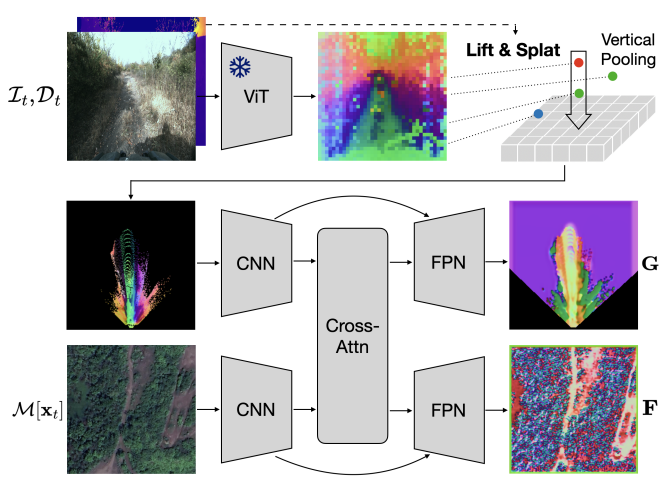


Fig. 3: BEV-Aerial feature network architecture.

We encode the ego-centric ground image $\mathcal{I}_t$ and local aerial image $\mathcal{M}[\mathbf{x}_t]$ into BEV features $\mathbf{G} \in \mathbb{R}^{H_g \times W_g \times D}$ and aerial feature map $\mathbf{F} \in \mathbb{R}^{H_a \times W_a \times D}$ through the BEV-aerial feature network. The on-board image is converted to BEV features so that it shares the same orthogonal, spatial view as the aerial map, enabling us to evaluate different pose hypotheses through feature matching. Figure 3 shows an overview of the network.

Given an image $\mathcal{I}_t$, the image encoder first extracts a feature map. We employ a pretrained visual-foundation model [15] to obtain features that are robust to appearance variations. Using the depth image $\mathcal{D}_t$, we lift the extracted features and splat them into 3-D space [16]. Next, max pooling is applied along the vertical axis, merging all features that fall within each BEV grid cell into a single representation. The BEV grid-cell size equals to the satellite map's spatial resolution (meters per pixel), ensuring a one-to-one correspondence between BEV cells and map pixels in the observation model.

The projected BEV embeddings and the local aerial image $\mathcal{M}[\mathbf{x}_t]$ are each passed through a ResNet-based CNN [4] followed by a Feature Pyramid Network (FPN) [12], yielding BEV features $\mathbf{G} \in \mathbb{R}^{H_g \times W_g \times D}$ and the aerial feature map $\mathbf{F} \in \mathbb{R}^{H_a \times W_a \times D}$. The output of last layer of CNN for BEV embeddings and aerial images are processed cross-attention module to fuse the information between BEV embeddings and aerial image. Neither $\mathbf{G}$ nor $\mathbf{F}$ is normalized, to let the model learn importance of feature by its magnitudes.

Training Objective. The BEV-aerial feature network is trained with supervised pairs of single RGB-D frames and ground-truth poses $\mathbf{x}^+$, together with negative samples $\mathcal{X}^-$ drawn around each ground-truth pose.

$$\mathcal{L} = -\sum_k \log \frac{\exp(\mathcal{S}(\mathbf{x}_k^+)/\tau)}{\exp(\mathcal{S}(\mathbf{x}_k^+)/\tau) + \sum_{\mathbf{x} \in \mathcal{X}_k^-} \exp(\mathcal{S}(\mathbf{x})/\tau)}. \quad (4)$$

For brevity, we write $\mathcal{S}(\mathbf{x}_t; \mathcal{I}_t, \mathcal{M})$ simply as $\mathcal{S}(\mathbf{x}_t)$, and $\tau \in \mathbb{R}^+$ is a scalar temperature parameter. The objective maximizes the similarity for the ground-truth pose while suppressing similarity for nearby negative poses, yielding a feature representation that is discriminative for metric localization.

IV. Experiments

We evaluate our method in off-road settings and seek to answer two questions: 1) **Tracking accuracy:** How precisely can the BEV-Patch-PF track the robot's pose? 2) **Generalization:** How robustly does the BEV-Patch-PF localize along routes that were never encountered during training?

A. Experimental Setup

Dataset. All experiments use the TartanDrive 2.0 off-road dataset [22], which provides stereo ground-level imagery, RGB images, GPS readings, and orientation data. High-resolution satellite orthophotos were obtained from an online imagery service and exported as GeoTIFFs in the appropriate UTM zone.[1] Depth images are computed from the stereo pairs with FoundationStereo [27]. Stereo visual odometry, later used as our motion model, was obtained with PyCuVSLAM [14]. We split the data into 28 training, 9 validation, and 22 test trajectories. The test set is further divided into *seen-routes* (6 trajectories overlapping the training paths) and *unseen-routes* (16 trajectories on partially or completely new tracks). Figure 4 illustrates the split.

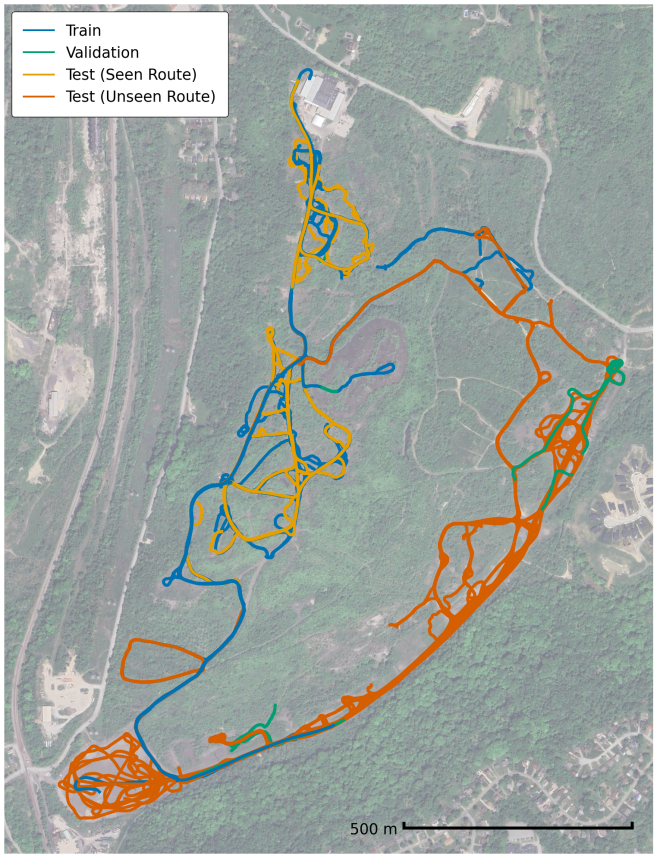
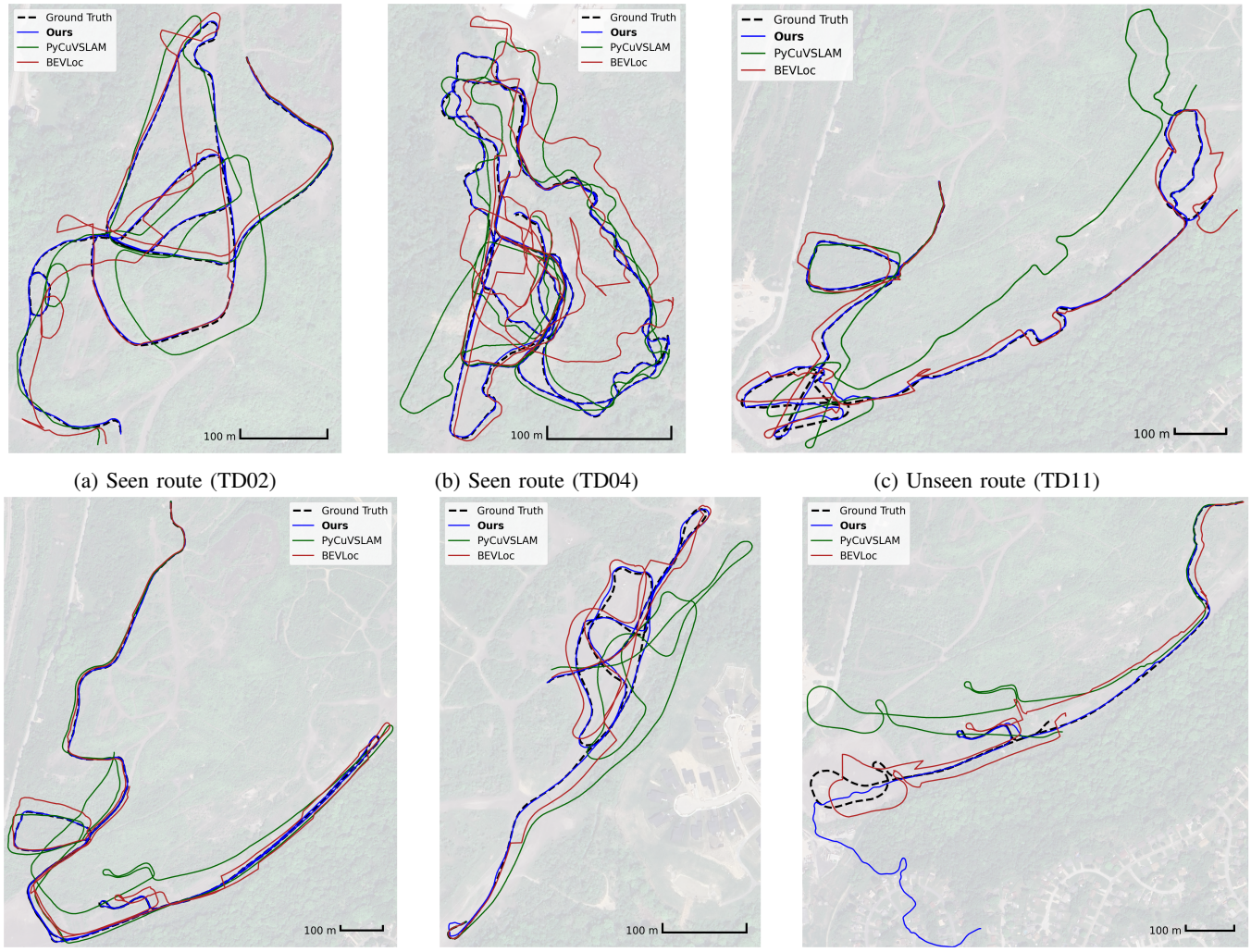


Fig. 4: Training, validation, and test splits for TartanDrive 2.0 [22]. Satellite imagery © 2025 Airbus, Maxar Technologies; map data © 2025 Google.

Baselines. We benchmark against two established alternatives: 1) PyCuVSLAM [14], a stereo visual odometry system; 2) BEVLoc [9]. We retrain the code released by the authors on our data split, replacing its original monocular TartanVO [26] with the more accurate stereo odometry from PyCuVSLAM. Following the authors' settings, we accumulate features over eight frames, update the pose prior with GPS every five iterations, and add registration factors to the pose-graph optimization only when the distance between the estimated pose and the GPS reading is below $300\,\mathrm{m}$. At every iteration we also set the orientation prior to the ground-truth orientation, as in their configuration.

Evaluation Metrics. We quantify performance using the Absolute Trajectory Error (ATE). Specifically, we compute the root-mean-square translational error between the estimated

[1] In our experiments, we loaded Google Satellite imagery into QGIS [17] and exported it as a GeoTIFF, reprojected to the target UTM zone 17 N.



(a) Seen route (TD02) (b) Seen route (TD04) (c) Unseen route (TD11)

(d) Unseen route (TD12) (e) Unseen route (TD15) (f) Failure case (TD16)

Fig. 5: Estimated trajectories produced by our model and by the baselines.

trajectory $\hat{\mathbf{T}}_{1:N}$ and the ground-truth trajectory $\mathbf{T}_{1:N}$:

$$\mathrm{ATE}_{\mathrm{RMS}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{p}_i - \hat{\mathbf{p}}_i\|^2},$$

with $\mathbf{p}_i$ and $\hat{\mathbf{p}}_i$ denoting the positions of $\mathbf{T}_i$ and $\hat{\mathbf{T}}_i$, respectively.

Implementation Details. Satellite imagery is downloaded at a resolution of $0.2\,\mathrm{m/pixel}$. During training, we randomly scale the aerial map within the range $0.18\,\mathrm{m/pixel}$ to $0.40\,\mathrm{m/pixel}$; all evaluations use $0.2\,\mathrm{m/pixel}$. The ground-image encoder is a frozen DINOv2 ViT-B/14 [15]. For BEV projection, we employ a voxel grid of size $(224, 224, 10)$, with a vertical resolution of $1\,\mathrm{m}$ and x–y size matched to the aerial map's spatial resolution. Both the projected BEV features and the local aerial map pass through a ResNet-18 backbone followed by two cross-attention layers, yielding 64-dimensional features. Input sizes are 518×518 for on-board images and 640×640 for local aerial images. During training, we sampled 63 negative poses uniformly within $\pm20m$ and $\pm60^\circ$ of the ground-truth pose.

For the particle-filter implementation, we use 128 particles. The filter is initialized around the ground-truth pose with Gaussian noise (σ_t=3 m translational, σ_θ=0.5 rad rotational). We used temperature of τ_s=0.5 for likelihood computation. Resampling is triggered when the effective sample size drops below 10% of the particles.

B. Experimental results

Tracking Accuracy. Our particle filter tracks *all seen trajectories* end-to-end, achieving a median $\mathrm{ATE}_{\mathrm{RMS}}$ of 1.30 m—significantly lower than PyCuVSLAM (23.74 m) and BEVLoc (16.81 m) (Tab. I). The absolute-pose-error CDF (Fig. 6) is strongly left-shifted, confirming the tighter error distribution. Representative trajectories are shown in Figs. 5a and 5b.

Generalization. On unseen routes, the particle filter maintains a median $\mathrm{ATE}_{\mathrm{RMS}}$ of 4.61 m, again outperforming

TABLE I: Absolute trajectory error (RMSE, meters) on TartanDrive 2.0 [22]. Scenes TD01–06 correspond to trajectories encountered during training, whereas TD07–22 correspond to unseen routes.

	Seen route (6 scenes)						Unseen route (5 scenes)				
Method	TD01	TD02	TD03	TD04	TD05	TD06	TD07	TD08	TD09	TD10	TD11
Ours	**1.71**	**1.52**	**0.86**	**0.95**	**1.11**	**1.65**	478.11	**1.46**	**3.36**	**12.63**	**5.36**
PyCuVSLAM [14]	10.91	22.67	79.02	15.67	10.87	3.32	269.29	15.94	144.09	28.13	121.86
BEVLoc [9]	16.15	24.78	17.07	33.84	5.97	3.07	**55.76**	23.75	16.64	17.23	22.69

	Unseen route (16 scenes, continued)										
Method	TD12	TD13	TD14	TD15	TD16	TD17	TD18	TD19	TD20	TD21	TD22
Ours	**2.12**	**10.71**	**2.22**	**3.87**	159.93	**1.59**	44.41	**1.98**	1044.61	234.63	**3.84**
PyCuVSLAM [14]	44.51	279.19	76.21	41.00	91.66	27.12	166.69	45.16	273.52	276.79	41.78
BEVLoc [9]	26.30	91.73	16.78	12.08	**35.38**	33.16	**23.87**	21.44	**27.05**	**25.53**	6.71

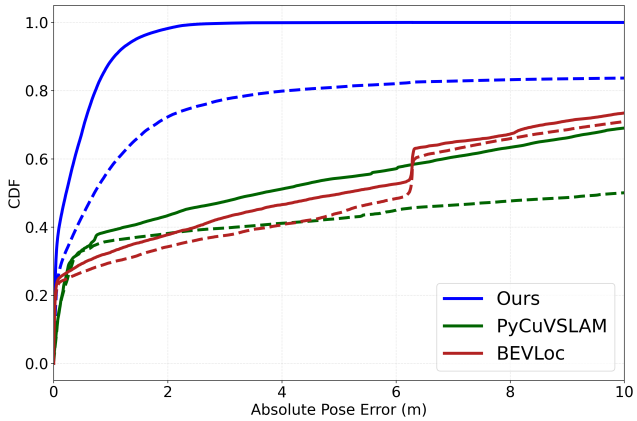


Fig. 6: Cumulative distribution of absolute pose error (solid: seen routes; dashed: unseen routes).

PyCuVSLAM (83.94 m) and BEVLoc (23.81 m) (Tab. I). The filter continuously corrects visual-odometry drift. BEVLoc [9] often produces jumpy, discontinuous poses because its pose-graph optimization cannot resolve ambiguous matching of per-frame localization. In contrast, once initialized, our filter stays smooth and accurate without any GPS data. Representative trajectories are shown in Figs. 5c, 5d, and 5e.

Failure Cases. The filter tracks reliably until it enters a large, feature-less open area. In such regions, our observation model assigns almost identical likelihoods to every particle. Because the weights are nearly flat, even a single false-positive match can tip the resampling step toward an arbitrary mode. Once the particles collapse onto that wrong hypothesis, later observations remain too ambiguous to pull the cloud back, so the filter never recovers. Figure 5f shows a representative failure segment.

V. Conclusion

This work introduced BEV-Patch-PF, a sequential cross-view geo-localization system that integrates a particle filter with a BEV-aerial feature-aligning observation model and a BEV-aerial feature network. By cropping a single aerial-feature patch for each pose hypothesis and matching it against BEV features, the system preserves accuracy while remaining memory-efficient. In real-world off-road experiments—both in revisited and unseen routes—BEV-Patch-PF consistently outperforms stereo visual odometry and a retrieval-based baseline, confirming the benefits of sequential inference and the learned BEV–aerial features.

However, the method still fails in feature-poor open fields, where the likelihood surface becomes nearly flat. To address these failure cases, we plan to estimate the distinctiveness of BEV feature and toggle the particle filter on or off to prevent catastrophic failures. For true zero-shot deployment across unseen regions and robot platforms, we further plan to: (i) train stronger, viewpoint-invariant descriptors so that identical terrain cues from multiple views converge in feature space; and (ii) eliminate the reliance on depth images to enable training on significantly larger datasets.

Acknowledgments

This work is partially supported by the ARL SARA (W911NF-24-2-0025). Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

References

[1] Siddharth Agarwal, Ankit Vora, Gaurav Pandey, Wayne Williams, Helen Kourous, and James McBride. Ford multi-av seasonal dataset. *The International Journal of Robotics Research*, 39(12):1367–1376, 2020.

[2] Francesco Castaldo, Amir Zamir, Roland Angst, Francesco Palmieri, and Silvio Savarese. Semantic cross-view matching. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 9–17, 2015.

[3] Florian Fervers, Sebastian Bullinger, Christoph Bodensteiner, Michael Arens, and Rainer Stiefelhagen. Uncertainty-aware vision-based metric cross-view geolocalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21621–21631, 2023.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In

Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[5] Sixing Hu and Gim Hee Lee. Image-based geo-localization using satellite imagery. *International Journal of Computer Vision*, 128(5):1205–1219, 2020.

[6] Sixing Hu, Mengdan Feng, Rang MH Nguyen, and Gim Hee Lee. 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*, pages 7258–7267, 2018.

[7] Lihong Jin, Wei Dong, Wenshan Wang, and Michael Kaess. Bevrender: Vision-based cross-view vehicle registration in off-road gnss-denied environment. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11032–11039. IEEE, 2024.

[8] Nikhil Keetha, Avneesh Mishra, Jay Karhade, Krishna Murthy Jatavallabhula, Sebastian Scherer, Madhava Krishna, and Sourav Garg. Anyloc: Towards universal visual place recognition. *IEEE Robotics and Automation Letters*, 9(2):1286–1293, 2023.

[9] Christopher Klammer and Michael Kaess. Bevloc: Cross-view localization and matching via birds-eye-view synthesis. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5656–5663. IEEE, 2024.

[10] Ted Lentsch, Zimin Xia, Holger Caesar, and Julian FP Kooij. Slicematch: Geometry-guided aggregation for cross-view pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17225–17234, 2023.

[11] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3292–3310, 2022.

[12] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[13] Feng Lu, Xiangyuan Lan, Lijun Zhang, Dongmei Jiang, Yaowei Wang, and Chun Yuan. Cricavpr: Cross-image correlation-aware representation learning for visual place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16772–16782, 2024.

[14] NVLabs. Pycuvslam. https://github.com/NVlabs/PyCuVSLAM, 2025. Accessed: 2025-04-25.

[15] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[16] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 194–210. Springer, 2020.

[17] QGIS Development Team. *QGIS Geographic Information System*. QGIS Association, 2025. URL https://www.qgis.org.

[18] Noe Samano, Mengjie Zhou, and Andrew Calway. You are here: Geolocation by embedding maps and images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pages 502–518. Springer, 2020.

[19] Paul-Edouard Sarlin, Daniel DeTone, Tsun-Yi Yang, Armen Avetisyan, Julian Straub, Tomasz Malisiewicz, Samuel Rota Bulo, Richard Newcombe, Peter Kontschieder, and Vasileios Balntas. Orienternet: Visual localization in 2d public maps with neural matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21632–21642, 2023.

[20] Yujiao Shi and Hongdong Li. Beyond cross-view image retrieval: Highly accurate vehicle localization using satellite image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17010–17020, 2022.

[21] Yujiao Shi, Fei Wu, Akhil Perincherry, Ankit Vora, and Hongdong Li. 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*, pages 21516–21526, 2023.

[22] Matthew Sivaprakasam, Parv Maheshwari, Mateo Guaman Castro, Samuel Triest, Micah Nye, Steve Willits, Andrew Saba, Wenshan Wang, and Sebastian Scherer. Tartandrive 2.0: More modalities and better infrastructure to further self-supervised learning research in off-road driving tasks. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12606–12606. IEEE, 2024.

[23] Zhenbo Song, Jianfeng Lu, Yujiao Shi, et al. Learning dense flow field for highly-accurate cross-view camera localization. *Advances in Neural Information Processing Systems*, 36:70612–70625, 2023.

[24] Yicong Tian, Chen Chen, and Mubarak Shah. Cross-view image matching for geo-localization in urban environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3608–3616, 2017.

[25] Aysim Toker, Qunjie Zhou, Maxim Maximov, and Laura Leal-Taixé. Coming down to earth: Satellite-to-street view synthesis for geo-localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6488–6497, 2021.

[26] Wenshan Wang, Yaoyu Hu, and Sebastian Scherer. Tartanvo: A generalizable learning-based vo. In *Conference on Robot Learning*, pages 1761–1772. PMLR, 2021.

[27] Bowen Wen, Matthew Trepte, Joseph Aribido, Jan Kautz, Orazio Gallo, and Stan Birchfield. Foundationstereo: Zero-shot stereo matching. *arXiv preprint arXiv:2501.09898*, 2025.

[28] Zimin Xia, Olaf Booij, Marco Manfredi, and Julian FP Kooij. Cross-view matching for vehicle localization by learning geographically local representations. *IEEE Robotics and Automation Letters*, 6(3):5921–5928, 2021.

[29] Zimin Xia, Olaf Booij, Marco Manfredi, and Julian FP Kooij. Visual cross-view metric localization with dense uncertainty estimates. In *European Conference on Computer Vision*, pages 90–106. Springer, 2022.

[30] Zimin Xia, Olaf Booij, and Julian FP Kooij. Convolutional cross-view pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3813–3831, 2023.

[31] Ali Younis and Erik Sudderth. Learning to be smooth: An end-to-end differentiable particle smoother. *Advances in Neural Information Processing Systems*, 37:7125–7155, 2024.

[32] Mengjie Zhou, Xieyuanli Chen, Noe Samano, Cyrill Stachniss, and Andrew Calway. Efficient localisation using images and openstreetmaps. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5507–5513. IEEE, 2021.

[33] Sijie Zhu, Taojiannan Yang, and Chen Chen. Vigor: Cross-view image geo-localization beyond one-to-one retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3640–3649, 2021.

[34] Sijie Zhu, Mubarak Shah, and Chen Chen. Transgeo: Transformer is all you need for cross-view image geo-localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1162–1171, 2022.