VGA: Empowering Aerial-Ground Localization by Visual Geometry Alignment

Supplementary Material

Overview

In this Supplementary Material, we provide the following:

1. Extended review of BEV and CVGL methods in Appendix A.
2. Extended implementation details in Appendix B.
3. Extended details on Training and Evaluation Dataset in Appendix C.
4. Additional Visualization examples of Calibration Prediction and BEV Geometry Prediction in Appendix D.
5. Related Downstream Tasks in Appendix E.
6. Failure Modes in Appendix F.

A. Extended Comparison to BEV and CVGL Methods

Existing BEV and CVGL methods [31, 58] primarily target 3D detection [22, 28, 35, 44, 46, 85], retrieval [40, 95], or geo-localization [15, 25, 49, 52, 56, 57, 59, 81–84, 93] between calibrated satellite imagery (with known ground sampling resolution and north alignment) and ground views, typically assuming a fixed camera height. These simplify the problem to estimating 3-DoF (x, y, θ) camera pose. Our task is fundamentally different: we address unconstrained 6-DoF relative pose estimation between aerial and ground views. Existing CVGL methods further assume a fixed camera height, simplifying the problem to a 3-DoF estimation(longitude, latitude, azimuth) and making them inapplicable to our setting, where no assumptions are made on camera height or calibration. In VGA, we instead use BEV projection and gravity prediction as geometric constraints to enable robust pose estimation under unknown pitch, roll, and scale. Therefore, existing BEV and CVGL methods are not directly applicable or comparable to our setting.

B. Extended Implementation Details

B.1. Neural BEV Projector

The Neural BEV Projector is designed for extracting and transforming information from encoded perspective image feature tokens T^a, T^g to learnable BEV queries Q^a, Q^g. It follows an encoder–decoder structure that employs modulated cross-attention to align image features with the BEV space conditioned on camera calibration, as illustrated in Fig. 5. Before each cross-attention operation, BEV queries Q are modulated by a Gated Feature-wise Linear Modulation layer [42]:

$$G(Q, c) = \gamma(\mathbf{c}) \odot [(1 + \alpha(\mathbf{c})) \odot Q + \beta(\mathbf{c})] \tag{18}$$

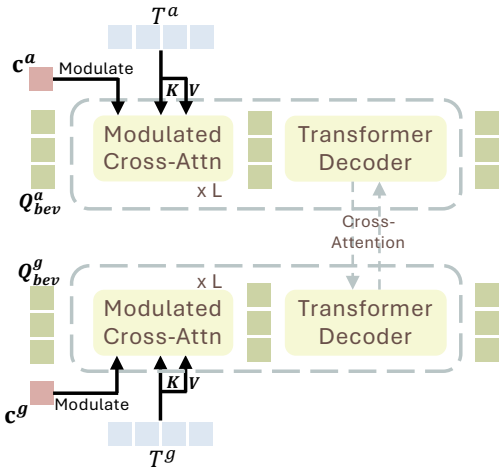

Figure 5. Overall Architecture of our Neural BEV Projector.

where $\mathbf{c}$ is the calibration token, $\alpha(\cdot), \beta(\cdot)$ and $\gamma(\cdot)$ are learnable affine projections and gating functions. This modulation ensures that BEV queries adapt dynamically to each camera's intrinsic and extrinsic configuration. Subsequent multi-head cross-attention allows interaction between the perspective features and BEV latent space, yielding calibration-aware BEV tokens that encode geometric context in the canonical, gravity-aligned frame. Finally, a cross-view BEV fusion decoder performs mutual cross-attention between aerial and ground BEV tokens, learning spatial correspondences and geometric consistency across views.

Architecture of BEV Projector. The Neural BEV Projector consists of 6 Modulated Cross-Attention Blocks and 6 Transformer Decoder Blocks. Each each attention layer is configured with a feature dimension of 1024 and employs 12 heads. We follow VGGT [72] to use QKNorm [16] and LayerScale [64] for each attention layer, We feed the BEV queries tokens output from 2-nd, 4-th and 6-th block into DPT for BEV prediction at size of 200×200 pixels, with 20×20 BEV query tokens.

B.2. Post-Geometry Optimization

As shown in Algorithm 1 and Algorithm 2, respectively, we present algorithm details for both Post-Geometry Optimization and RANSAC-Weighted Procrustes Alignment introduced in Sec. 3.5. We further ablate the choice of weights for Post-Geometry Optimization in Fig. 6. During inference, we fix $\lambda_S = 1 \times 10^{-2}$ and vary λ_α and λ_g to search for best combination of weights.

Algorithm 1 Post-Geometry Optimization

Require: Initial relative pose (R_0, t_0), inlier correspondence $\{(X_i^a, X_i^g)\}$, camera intrinsics (K_1, K_2), mask $\mathcal{M}$, gravity vectors $(\vec{\mathbf{g}}_1, \vec{\mathbf{g}}_2)$, Azimuth prior α, weights $\lambda_g, \lambda_\alpha, \lambda_\text{S}$, iterations T.
Ensure: Refined relative pose (R^*, t^*).
Normalize pixel coordinates: $\mathbf{f}_1 \leftarrow K_1^{-1} X^a$, $\mathbf{f}_2 \leftarrow K_2^{-1} X^g$
Initialize $R \leftarrow R_0$, $t \leftarrow \frac{t_0}{\|t_0\|}$
for $i = 1 \dots T$ **do**
Compute essential matrix: $E = [t]_\times R$
Compute Sampson residuals r_s from $(E, \mathbf{f}_1, \mathbf{f}_2)$
Compute gravity residual:
$r_g = \sqrt{\lambda_g} \arccos\big((R\vec{\mathbf{g}}_1)^\top \vec{\mathbf{g}}_2\big)$
Compute yaw residual using global up vector $\mathbf{u}$:
$r_\alpha = \sqrt{\lambda_\alpha} \| \log({R_{z,0}}^T R_z(\alpha)) \|$
Stack residuals $r = [r_s, r_g, r_\alpha]$
Estimate Jacobian J of r w.r.t. pose increments $(\delta R, \delta t)$
Solve $(J^\top J + \lambda I)\Delta = -J^\top r$
Update $R \leftarrow R \exp([\Delta_R]_\times)$, $t \leftarrow \frac{t + \Delta_t}{\|t + \Delta_t\|}$
end for
return $(R^*, t^*) \leftarrow (R, t)$

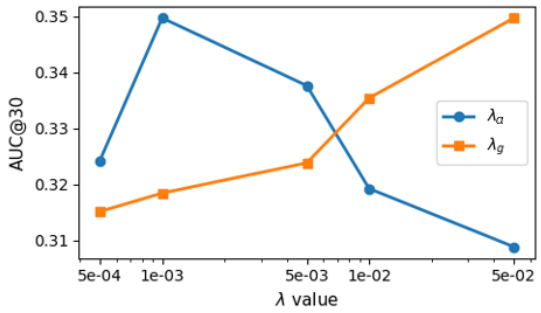

Figure 6. Ablation on gravity and azimuth weighting on MatrixCity(BigCity) benchmark. AUC@30 performance of varying loss weights λ_g with fixed $\lambda_\alpha = 1e^{-3}$, and λ_α with fixed $\lambda_\text{g} = 5e^{-2}$.

Efficiency and Memory Consumption. The inference GPU Memory usage of our model running with FP16 requires about 10.56GB for each pair. The inference time required for each component are as following:

- Model feed-forward: ∼550ms, similar to MASt3R, which takes around 500ms,
- LM Optimization for calibration parameters: ∼270ms,
- Post-Geometry Joint optimization: ∼30ms,

tested on a workstation with AMD Ryzen 9 3900X 12-Core Processor and NVIDIA GeForce RTX 3090. Overall, the two optimization steps introduce only a small additional overhead relative to the forward pass, but substantially improving the final 6-DoF pose accuracy. This demonstrates that our inference-time refinement remains lightweight compared to the core model inference, yielding significant performance gains at the same time.

Algorithm 2 RANSAC-Weighted Procrustes Alignment

Require: BEV correspondences $\{(\mathbf{a}_i, \mathbf{b}_i)\}_{i=1}^N$, weights w_i, inlier threshold τ, iterations N.
Ensure: Estimated similarity transform (s^*, R^*, t^*).
Initialize best inlier count $n^* \leftarrow 0$
for $k = 1 \dots N_\text{it}$ **do**
Randomly sample a minimal subset $\mathcal{S} \subset \{1, \dots, N\}$
Normalize weights: $\tilde{w}_i = \frac{w_i}{\sum_{j \in \mathcal{S}} w_j}$, $i \in \mathcal{S}$
Compute centroids: $\bar{\mathbf{a}} = \sum_{i \in \mathcal{S}} \tilde{w}_i \mathbf{a}_i$, $\bar{\mathbf{b}} = \sum_{i \in \mathcal{S}} \tilde{w}_i \mathbf{b}_i$
Center points: $\mathbf{a}'_i = \mathbf{a}_i - \bar{\mathbf{a}}$, $\mathbf{b}'_i = \mathbf{b}_i - \bar{\mathbf{b}}$
Form weighted covariance: $H = \sum_{i \in \mathcal{S}} \tilde{w}_i \, \mathbf{a}'_i {\mathbf{b}'_i}^\top$
Compute SVD: $H = U \Sigma V^\top$
Set $R_k = V U^\top$
Compute scale: $s_k = \frac{\text{trace}(\Sigma)}{\sum_{i \in \mathcal{S}} \tilde{w}_i \|\mathbf{a}'_i\|^2}$
Compute translation: $t_k = \bar{\mathbf{b}} - s_k R_k \bar{\mathbf{a}}$
For all i, compute residual $r_i = \big\|\mathbf{b}_i - (s_k R_k \mathbf{a}_i + t_k)\big\|_2$
Define inlier set $\mathcal{I}_k = \{i \mid r_i < \tau\}$
if $|\mathcal{I}_k| > n^*$ **then**
$n^* \leftarrow |\mathcal{I}_k|$
$(s^*, R^*, t^*) \leftarrow (s_k, R_k, t_k)$
$\mathcal{I}^* \leftarrow \mathcal{I}_k$
end if
end for
return (s^*, R^*, t^*)

C. Extended Details on Dataset

Training and Evaluation Dataset Curation. We curate both MatrixCity (SmallCity) and AerialMegaDepth for training our model with BEV supervision, as illustrated in Fig. 7. Our goal is to select aerial–ground image pairs that exhibit sufficient overlap in both the perspective and BEV domains, enabling effective supervision of BEV geometry and cross-view matching. Given ground-truth depth, we follow MASt3R [26] to compute pairwise 3D correspondences via nearest-neighbor search using K-d trees, and retain pairs containing more than 30 matched BEV pixels on the projected BEV plane. For AerialMegaDepth, we perform gravity alignment before curation using COLMAP [50], applying Manhattan-world alignment to orient the global y-axis along gravity.

For evaluation datasets lacking ground-truth depth, we identify valid aerial–ground pairs through geometric frustum filtering. Specifically, each aerial image's camera frus-

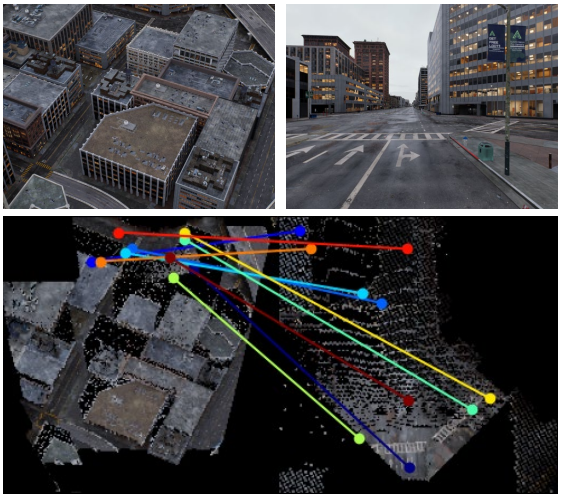

Figure 7. Data Curation for our BEV supervision. Given aerial-ground Image pair, we construct our ground-truth Union BEV Pointmap and BEV correspondences for supervision as introduced in Sec. 3.4.

tum is projected into the ground-view coordinate frame, and candidate ground images are retained if their camera centers fall within the aerial frustum and exhibit sufficient view-overlap ratio. This procedure ensures that evaluation pairs maintain meaningful spatial correlation and comparable scene coverage, even in the absence of dense depth supervision. Such a strategy allows fair benchmarking of localization performance under realistic conditions where only sparse or no 3D information is available.

D. Additional Qualitative Results

In Fig. 8, we present additional camera localization result by our method, we also provide reconstructed 3D point cloud glb files.

Perspective Field Prediction in the Wild. In Fig. 9, we visualize our predicted perspective fields on aerial–ground image pairs from AerialMegaDepth [69], which provides in-the-wild aerial-ground pairs without ground-truth camera poses. Column 2 shows our raw perspective field predictions, while column 3 illustrates the optimized results after LM optimization. For comparison, column 4 presents the outputs of GeoCalib [67], a state-of-the-art monocular camera calibration method. Our model demonstrates significantly improved calibration on aerial images captured with large pitch and roll angles, where GeoCalib often fails to recover true gravity direction, as 0 degree latitude should not appear on near-nadir aerial imagery. This improvement highlights the robustness of our learned perspective field representation, which benefits from the joint aerial–ground training and the geometric coupling between the two views. Notably, our calibration refinement produces coherent gravity across both views, offering a physically interpretable intermediate representation that directly supports downstream tasks such as gravity-aligned BEV projection and multi-view pose optimization.

Perspective Field Prediction vs. Ground-truth. In Fig. 10, we compare our raw perspective field predictions (column 4) and our optimized estimation (column 3) against ground-truth perspective fields from the MatrixCity (BigCity) and ACC-NVS1 benchmarks. Across both datasets, our method accurately recovers gravity direction and vertical field-of-view, demonstrating strong zero-shot generalization. The optimized estimates consistently reduce residual error of gravity and latitude prediction, indicating that our calibration refinement effectively enforces global geometric consistency. These results validate the reliability of our learned perspective field representation and highlight its critical role in downstream pose refinement and BEV alignment.

E. Related Downstream Tasks

Beyond pose estimation, VGA explicitly predicts camera calibration, BEV projection geometry, and cross-view alignment. These explicit geometric outputs of VGA directly enable aerial-ground keypoint matching and tracking-Fig. 11, cross-view 3D reconstruction, and BEV-based 3D object detection and tracking. We focus on localization as a representative evaluation task, since reliable gravity-aligned pose estimation is a prerequisite for these downstream applications. illustrative examples are provided in the supplementary material.

F. Failure Modes

Similar to other correspondence-based approaches, VGA can fail when shared geometric constraints between views become insufficient. We identify three primary failure modes: (1) *Extremely low visual overlap* (typically below 10%) or severe occlusion, where reliable pose estimation via PnP becomes infeasible regardless of geometric prior quality; (2) *Scenes lacking dominant planar structures or salient landmarks*: such as environments dominated by unstructured vegetation or landscape, where planar alignment priors are ineffective; and (3) *Severe scale or altitude mismatch* (exceeding $\sim$500m altitude difference), which destabilizes the projection geometry and BEV correspondences. Gravity prediction itself fails only in rare cases where gravity orientation is visually unobservable (e.g., near-nadir imagery without architectural cues). A representative failure example is illustrated in Fig. 12, which all other methods failed too.

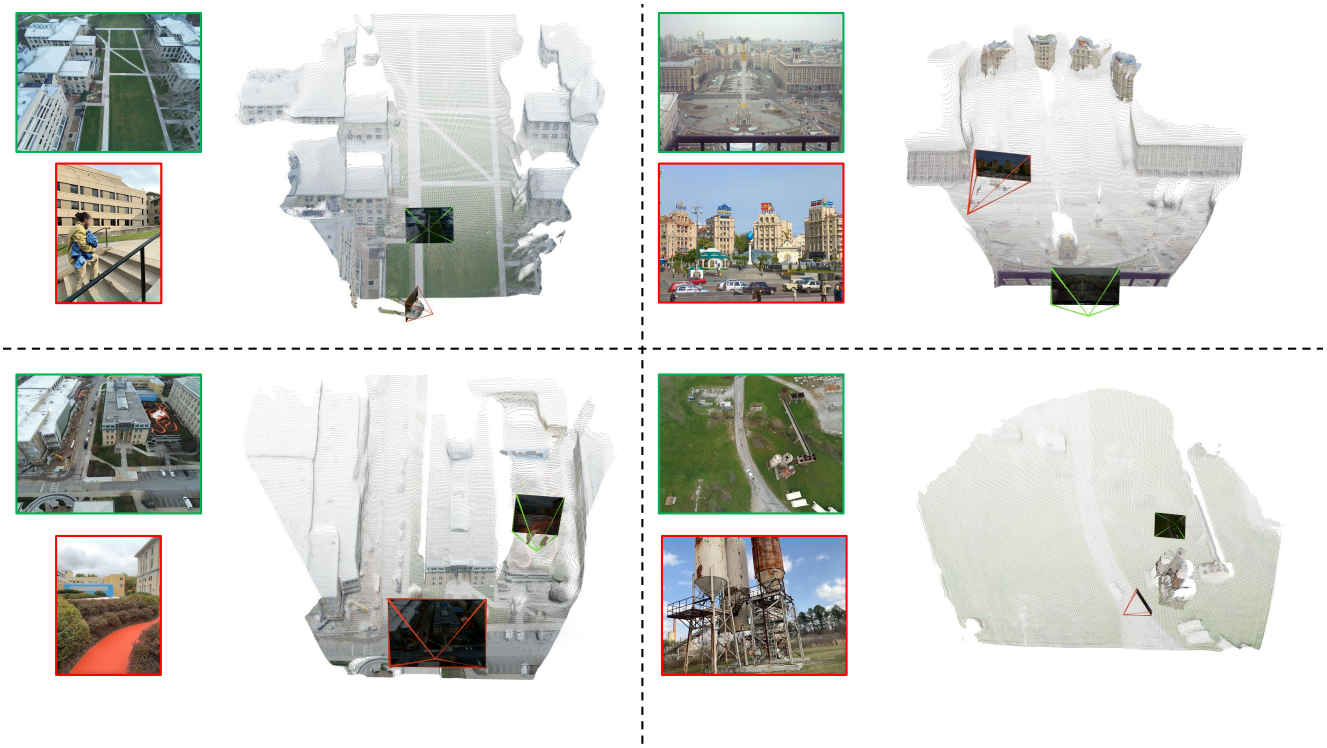

Figure 8. Additional visualization of Camera Localization and reconstruction results. Best viewed zoomed in.

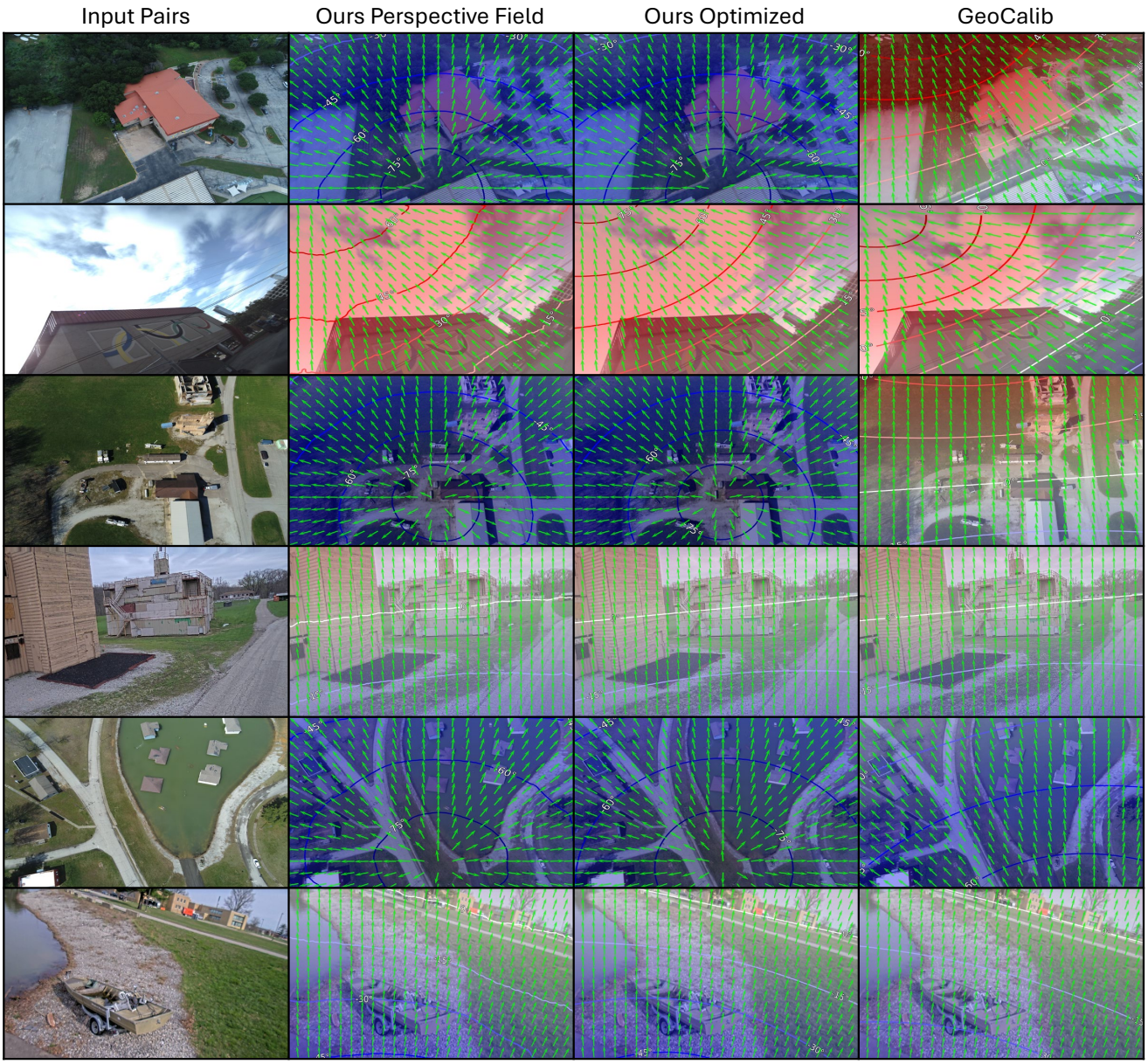

Figure 9. Visualization of Perspective Field Prediction for Aerial-Ground Pairs in the wild. We highlight the failure cases in the aerial views predicted by GeoCalib.

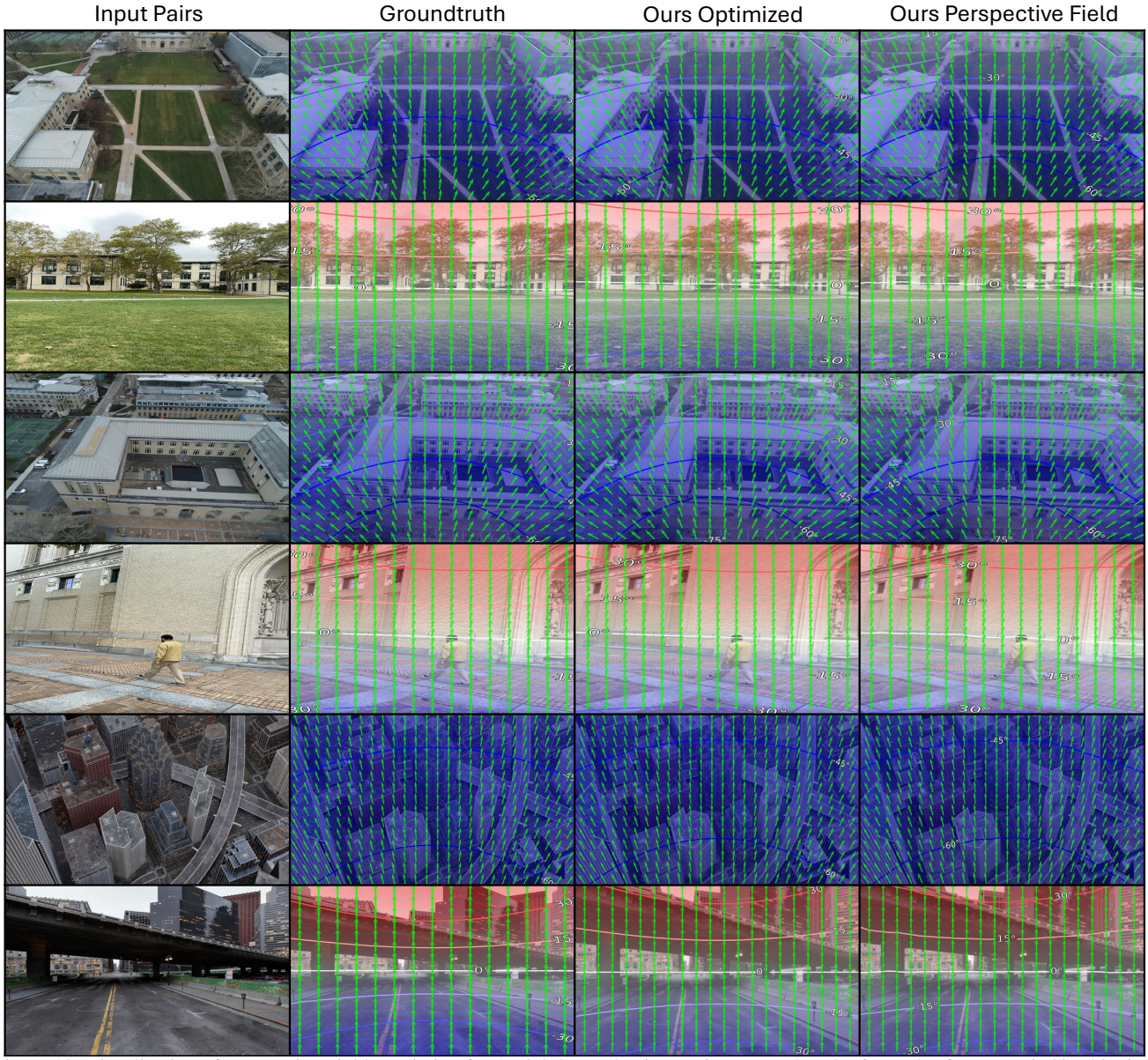

Figure 10. Visualization of Perspective Field Prediction for Aerial-Ground Pairs, against ground truth of samples from MatrixCity(bigcity) and ACC-NVS1. We highlight the accurate Perspective Field prediction, leading to accurate gravity and camera intrinsics prediction.

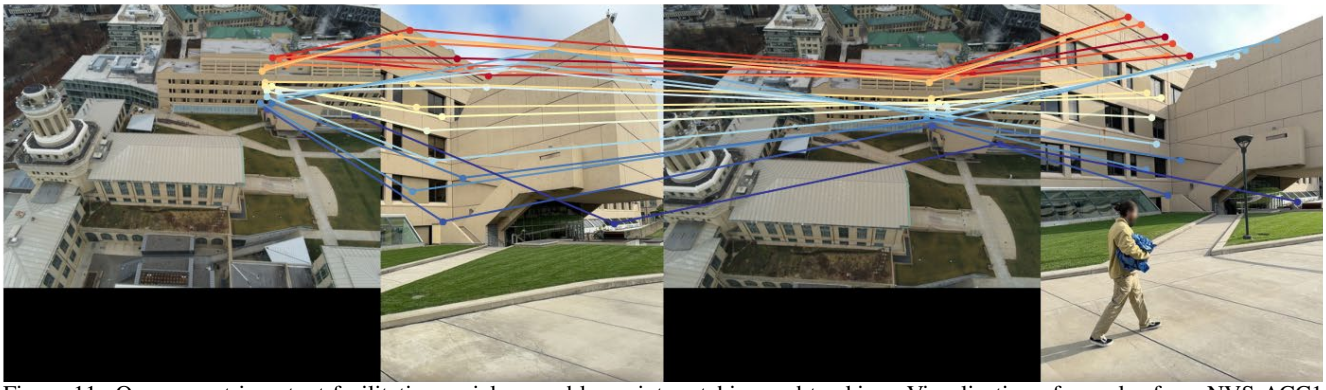

Figure 11. Our geometric output facilitating aerial-ground keypoint matching and tracking. Visualization of samples from NVS-ACC1 benchmark. Best viewed zoom in.

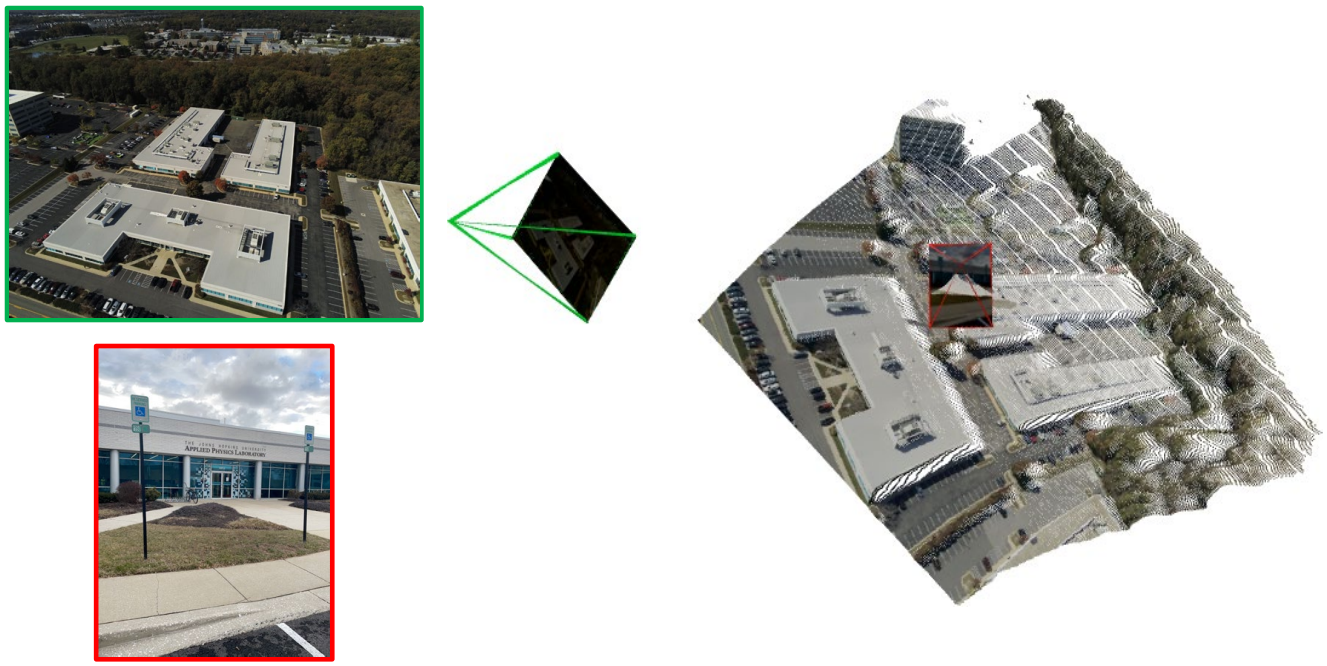

Figure 12. Example of failure case, caused by severe scene altitude/scale difference and extremely low visual overlap. As a result, the model fails to localize correct camera pose due to lack of reliable correspondent matches.