FG^2: Fine-Grained Cross-View Localization by Fine-Grained Feature Matching

Supplementary Material

In this supplementary material, we provide the following information to support the main paper:

A Projection geometry.
B Procrustes alignment.
C Additional details about RANSAC.
D Additional comparison of loss functions.
E Assumption on orthographic images.
F Directly matching DINO feature.
G Extra Qualitative Results.

A. Projection geometry

We provide details on how to find the projected pixel coordinates of the 3D points in the ground image feature map $f(G)$ during the mapping to 3D step in Sec. 3.2 of the main paper.

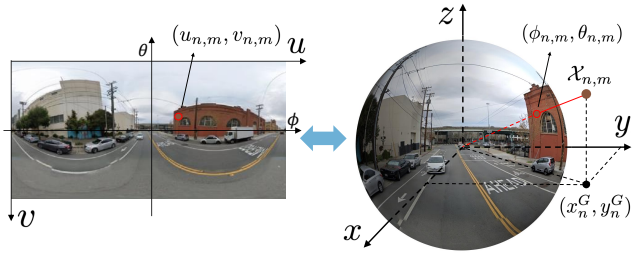

Figure 1. Projecting a 3D point to a panoramic image. We use the original image for visualization purposes. In practice, we find the projected pixel coordinates in the extracted feature map.

We begin with the case where ground images are panoramic. As shown in Fig. 1, a panorama represents the surface of a sphere. Each pixel in a panorama is associated with a spherical coordinate (ϕ, θ), where $\phi \in [-\pi, \pi]$ and $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. Given the coordinate $(x_n^G, y_n^G, z_{n,m}^G)$ of the 3D point $\mathcal{X}_{n,m}$, where n is the index of the ground 2D point in the created point set ξ^G and m is the index of the lifted 3D point, we compute its projected spherical coordinates $(\phi_{n,m}, \theta_{n,m})$ using,

$$\phi_{n,m} = \begin{cases} \arccos(\frac{x_n^G}{\sqrt{(x_n^G)^2+(y_n^G)^2}}), & \text{if } y_n^G \geq 0 \\ -\arccos(\frac{x_n^G}{\sqrt{(x_n^G)^2+(y_n^G)^2}}), & \text{otherwise,} \end{cases} \quad (1)$$

$$\theta_{n,m} = \arcsin(\frac{z_{n,m}^G}{\sqrt{(x_n^G)^2 + (y_n^G)^2 + (z_{n,m}^G)^2}}). \quad (2)$$

Once we have $(\phi_{n,m}, \theta_{n,m})$, we can simply find its corresponding pixel coordinates $(u_{n,m}, v_{n,m})$,

$$u_{n,m} = \frac{\phi_{n,m} + \pi}{2\pi} W, \quad (3)$$

$$v_{n,m} = \frac{\frac{\pi}{2} - \theta_{n,m}}{\pi} H, \quad (4)$$

where W and H are the width and height of the extracted ground feature map $f(G)$.

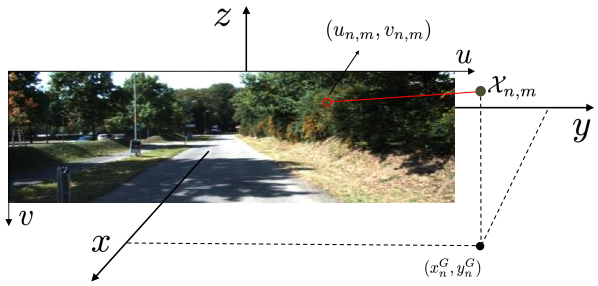

Figure 2. Perspective projection of a 3D point. We use the original image for visualization purposes. In practice, we find the projected pixel coordinates in the extracted feature map.

If the ground images follow a perspective projection, see Fig. 2, the pixel coordinates $(u_{n,m}, v_{n,m})$ of each 3D point $\mathcal{X}_{n,m} = (x_n^G, y_n^G, z_{n,m}^G)$ can be computed using the camera intrinsics,

$$u_{n,m} = \frac{W}{2} + \frac{\text{focal length} \cdot y_n^G}{x_n^G \cdot \text{stride}} + u_0, \quad (5)$$

$$v_{n,m} = \frac{H}{2} - \frac{\text{focal length} \cdot z_{n,m}^G}{x_n^G \cdot \text{stride}} + v_0, \quad (6)$$

where stride is the downscale factor between the input image G and its extracted feature map $f(G)$ and (u_0, v_0) denotes the amount of principal point offset in the feature map $f(G)$.

B. Procrustes alignment

As mentioned in Sec. 3.3 of the main paper, we use the Kabsch solver [5], also known as the orthogonal Procrustes algorithm, to compute the relative pose between ground and aerial 2D point sets ξ^G and ξ^A. Kabsch solver is common for point cloud registration.

The Kabsch solver computes the optimal rotation matrix that minimizes the root mean squared deviation between two paired sets of points. The translation between the two point sets is determined by calculating the shift between the

resulting two rotation-aligned point sets. This process is inherently differentiable [2, 3], as the rotation is obtained through the Singular Value Decomposition (SVD) of the cross-correlation matrix between the two point sets.

As noted in Sec. 3.3, given the N ground and aerial 2D points ξ^G and ξ^A and their pair-wise matching probability D (of length of N^2), we sample N_S correspondences,

$$\{D_{n_S}\} \sim D, \quad n_S = 1, ..., N_S, \tag{7}$$

where n_S is the sampled index in D. We denote the ground and aerial points of those N_S correspondences as ξ_S^G and ξ_S^A. Then, the Kabsch solver computes the rotation (in our case, yaw orientation o) between ξ_S^G and ξ_S^A in three steps.

First, ξ_S^G and ξ_S^A are translated such that their centroid coincides with the origin of the coordinate system,

$$\xi_{\text{trans}}^G = \xi_S^G - \text{mean}(\xi_S^G), \quad \xi_{\text{trans}}^A = \xi_S^A - \text{mean}(\xi_S^A), \tag{8}$$

where ξ_{trans}^G and ξ_{trans}^A represent the translated ground and aerial points, respectively, and mean$(\cdot)$ computes the mean coordinate of the points in ξ_S^G and ξ_S^A. When computing the mean, the matching probabilities $\{D_{n_S}\}$ are used as weights [4], which means a point with a higher matching probability contributes more to the mean coordinate.

Then, we compute a cross-covariance matrix H between ξ_{trans}^G and ξ_{trans}^A as $H = (\xi_{\text{trans}}^A)^T \cdot \xi_{\text{trans}}^G$.

Finally, we compute the SVD of H,

$$U \cdot S \cdot V^T = H, \tag{9}$$

where U and V are orthogonal and S is diagonal. The 2×2 rotation matrix rot of the orientation o is then,

$$rot = U \cdot \begin{bmatrix} 1 & 0 \\ 0 & \text{determinant}(U \cdot V^T) \end{bmatrix} \cdot V^T. \tag{10}$$

Once the orientation is estimated, the translation $\boldsymbol{t}_m$ between ξ_S^G and ξ_S^A is computed as $\boldsymbol{t}_m = \text{mean}(\xi_S^G) - \text{mean}(\xi_S^A) \cdot rot^T$.

C. Additional details about RANSAC

RANSAC is not used during training. When inference with RANSAC, we conduct $R = 100$ iterations and use 2.5 m as the threshold in Eq. 5 in the main paper. As shown in Tab. 1, on the VIGOR same-area test set, RANSAC reduces the mean localization error from 2.18 m to 1.95 m. Without RANSAC, our method achieves 9.26 FPS on an H100 GPU. As expected, RANSAC increases computation, and inference with RANSAC has 0.32 FPS. If fast runtime is required, our method can be used without RANSAC, and it still surpasses the previous state-of-the-art in localization accuracy. RANSAC is used to obtain the accurate pose estimation reported in the main paper (Tab. 1), whereas it is omitted in the ablation study (Tab. 3) due to runtime considerations. On the KITTI dataset, we did not observe an overall performance improvement with RANSAC. Therefore, we did not use RANSAC on KITTI.

VIGOR	FPS	Same-area		Cross-area	
		Mean	Median	Mean	Median
w/o. R.	**9.26**	2.18	1.18	2.74	1.52
w. R.	0.32	**1.95**	**1.08**	**2.41**	**1.37**

Table 1. Comparison of localization errors and runtime between our model with ("w. R.") and without RANSAC ("w/o. R.") when performing inference on the VIGOR test set with known orientation. **Best in bold.**

D. Additional comparison of loss functions

We use a Virtual Correspondence Error loss, $\mathcal{L}_{VCE}$, adapted from [1], to supervise the camera pose (main paper Sec. 3.4). Here, we also compare it with other pose losses.

We tested a loss function that sums an L1 loss for localization and another L1 loss for orientation estimation, denoted as $\mathcal{L}_A$, and an alternative, $\mathcal{L}_B$, which replaces the L1 loss for localization with an L2 loss (i.e., minimizing squared distance). As shown in Tab. 2, both $\mathcal{L}_A$ and $\mathcal{L}_B$ performed worse than our $\mathcal{L}_{VCE}$. $\mathcal{L}_{VCE}$ jointly considers localization and orientation, potentially leading to a better local optimum during training.

Error (m)	$\mathcal{L}_{VCE}$	$\mathcal{L}_A$	$\mathcal{L}_B$
Mean	**2.17**	2.68	2.58
Median	**1.18**	1.38	1.37

Table 2. Loss comparison on VIGOR validation set with known orientation. **Best in bold.**

E. Assumption on orthographic images

Our method assumes each pixel in the aerial image corresponds to a vertical ray in 3D (main paper Sec. 3.1). However, existing cross-view localization datasets often violate this assumption.

We do not explicitly address non-orthographic aerial images. Since the model is supervised using the camera pose, or equivalently, the ground and aerial BEV point locations, it must learn to assign matchable features to corresponding ground and aerial points. During training, building facades are sometimes visible in the aerial image, while at other times only the rooftops are seen. As a result, the model must learn to associate ground-view content (e.g., buildings) with both facades and roofs. This likely explains why, in the aerial view, points on buildings in the ground view are

sometimes matched to roofs (main paper Fig. 3c) and other times to facades (main paper Fig. 3a).

On the other hand, since our point set is relatively sparse and the point descriptors are constructed from deep features, these descriptors likely summarize information from local neighborhoods. It is also possible that the descriptors for roof or facade points capture more global semantic information, e.g., identifying a region as part of a building, rather than strictly encoding the visual appearance of the projected location in the input images.

F. Directly matching DINO feature

As mentioned in the main paper, we use pre-trained DINOv2 features [6]. We tried directly matching DINO features between ground and aerial images. However, this approach did not produce visually reasonable matches, see Fig. 3. Therefore, using our model on top of the DINO features is essential.

The DINOv2 paper [6] demonstrated impressive results by directly matching features extracted from images and sketches. However, ground-to-aerial cross-view image matching is significantly more challenging due to factors such as the lack of clear separation between foreground and background, the presence of dynamic and unmatchable objects, the repetition of structures like buildings and trees, and the substantial differences in perspective, scale, and scene coverage between the two views.

G. Extra qualitative results

Finally, in addition to the qualitative results presented in Sec. 4.4 of the main paper, we provide more examples in Fig. 4 and 5. In Fig. 4 (a)-(d), we include the same samples as in Fig. 3, and our method produces reasonable matches. In Fig. 5 (i)-(l), we show predictions with unknown orientation.

References

[1] Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Aron Monszpart, Victor Prisacariu, Daniyar Turmukhambetov, and Eric Brachmann. Map-free visual relocalization: Metric pose relative to a single image. In *European Conference on Computer Vision*, pages 690–708. Springer, 2022. 2

[2] Armen Avetisyan, Angela Dai, and Matthias Nießner. End-to-end cad model retrieval and 9dof alignment in 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2551–2560, 2019. 2

[3] Eric Brachmann and Carsten Rother. Visual camera relocalization from rgb and rgb-d images using dsac. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5847–5865, 2021. 2

[4] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2514–2523, 2020. 2

[5] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976. 1

[6] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. 3, 4

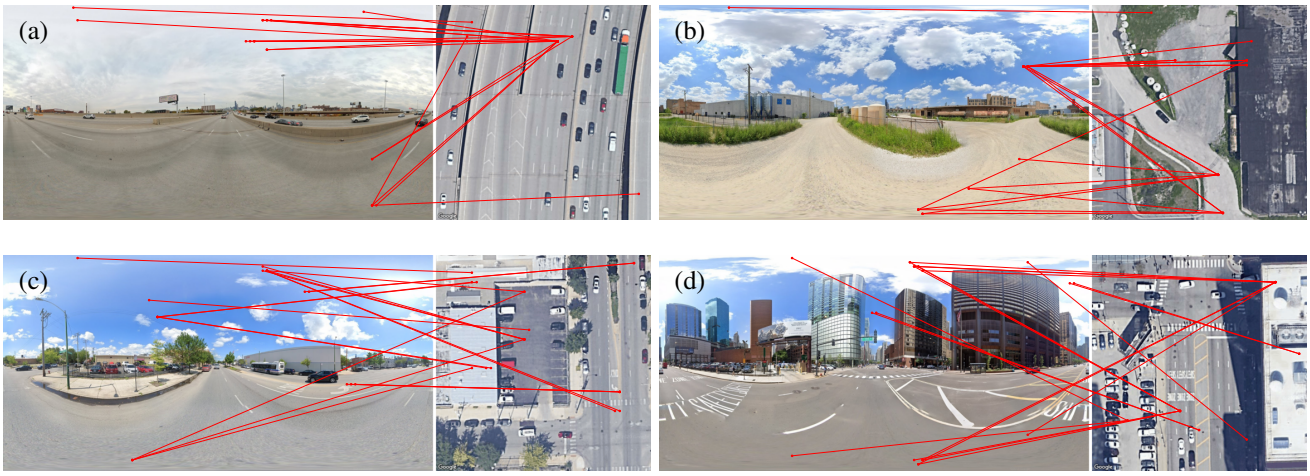

Figure 3. Directly matching DINOv2 features [6] of the ground and aerial images. We show the 20 matches with the highest similarity scores.

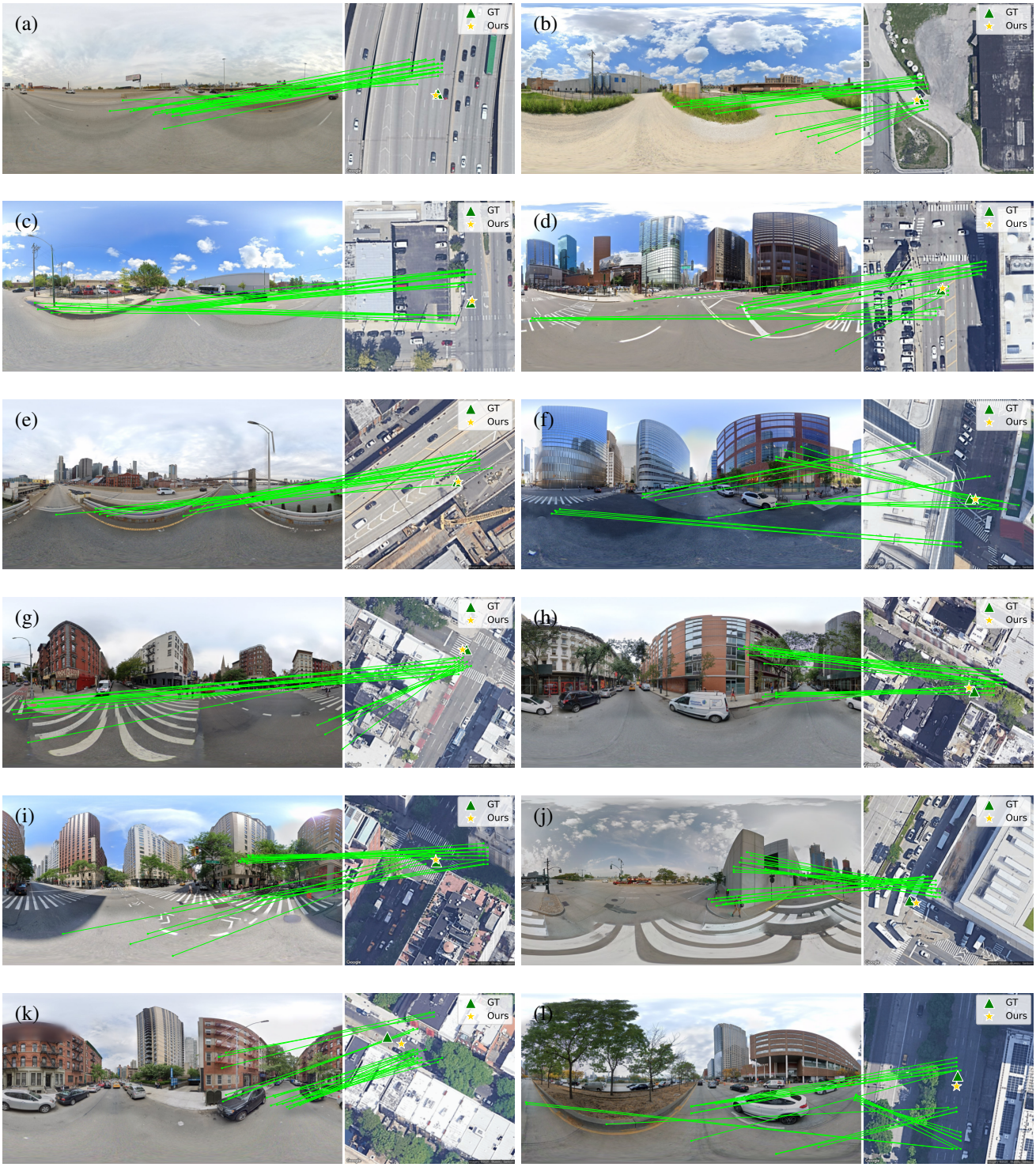

Figure 4. Fine-grained feature matching results on VIGOR with known orientation. We show the 20 matches with the highest similarity scores. We find the 3D points using the selected height in the last pooling to BEV step and then project those points to the ground image.

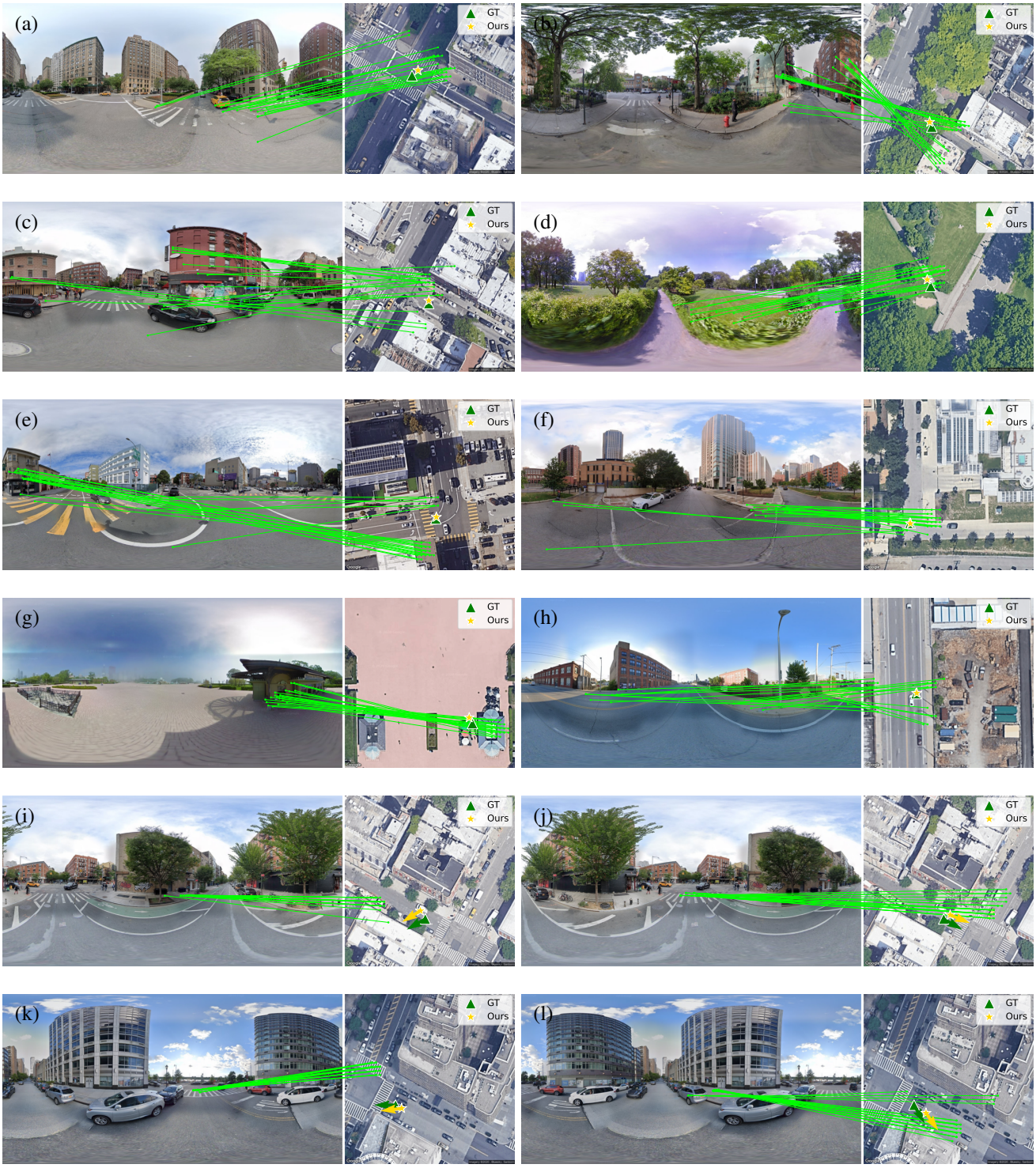

Figure 5. Fine-grained feature matching results on VIGOR with known and unknown orientation. We show the 20 matches with the highest similarity scores. We find the 3D points using the selected height in the last pooling to BEV step and then project those points to the ground image.