This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

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

Zimin Xia Alexandre Alahi
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
{zimin.xia,alexandre.alahi}@epfl.ch

Abstract

We propose a novel fine-grained cross-view localization method that estimates the 3 Degrees of Freedom pose of a ground-level image in an aerial image of the surroundings by matching fine-grained features between the two images. The pose is estimated by aligning a point plane generated from the ground image with a point plane sampled from the aerial image. To generate the ground points, we first map ground image features to a 3D point cloud. Our method then learns to select features along the height dimension to pool the 3D points to a Bird's-Eye-View (BEV) plane. This selection enables us to trace which feature in the ground image contributes to the BEV representation. Next, we sample a set of sparse matches from computed point correspondences between the two point planes and compute their relative pose using Procrustes alignment. Compared to the previous state-of-the-art, our method reduces the mean localization error by 28% on the VIGOR cross-area test set. Qualitative results show that our method learns semantically consistent matches across ground and aerial views through weakly supervised learning from the camera pose.

1. Introduction

Visual localization aims to estimate the camera pose of a given image relative to a representation of a scene. For large-scale applications like autonomous driving and outdoor mobile robotics, the scalability of this representation is crucial, as mapping is expensive [50]. Recently, using aerial imagery for accurate visual localization [20, 29, 33–36, 39, 42, 43, 46–48] has gained increasing interest due to its global coverage. This task, named fine-grained cross-view localization, aims to estimate the 3 Degrees of Freedom (DoF) pose, i.e., the planar location and yaw orientation, of a query ground-level image on an aerial image that covers the ground camera's local surroundings. In practice, the aerial image can be identified using noisy position estimates from onboard Global Navigation Satellite System (GNSS) in vehicles or robots, which contain errors up to tens of meters in urban canyons [4].

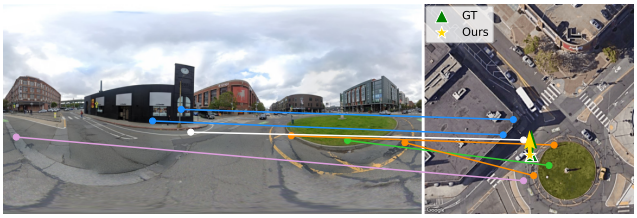


Figure 1. Fine-grained cross-view localization estimates the 3 DoF pose of a ground image (left) on an aerial image (right) of the surroundings. Our method tackles this task by matching fine-grained local features across views, providing interpretable results. The matched correspondences are semantically consistent and learned *without ground truth labels*. Here are a few selected predictions.

In fact, fine-grained cross-view localization is common in everyday life, such as when people use Google Satellite Maps [13] to determine their position. In this process, people first identify surrounding landmarks, such as buildings and lane markings, understand their relative positions, and then compare them to the aerial image. However, existing methods do not fully follow this intuitive approach.

Global descriptor-based methods [20, 46, 47] generate descriptors at candidate locations in the aerial image and compare them to the descriptor of the query ground image. Intuitively, this approach mimics imagining what would be visible in the ground view at each candidate location given the aerial view but is not how humans naturally approach localization. Local feature-based methods typically transform the ground image into a Bird's-Eye-View (BEV) representation using homography [36, 39]. However, homography applies only to the ground plane, limiting these methods by overlooking above-ground objects, such as buildings. Besides, some methods [9, 34] estimate the ground camera's pose by densely comparing the ground view BEV representation to the aerial view. Yet humans tend to localize based on a few distinct objects [7, 44], rather than the entire panoramic scene. Moreover, no existing method explicitly identifies which objects in the ground view match those in the aerial view, making localization results hard to interpret.

We bridge the above gaps by proposing a novel fine-grained cross-view localization method that aligns with hu-

man intuition. As shown in Fig. 1, our method matches fine-grained local features across views and solves localization through sparse correspondences. Similar to prior works, we also transform the ground-level image into a BEV representation. However, our method is unique such that it first explicitly generates a 3D point cloud centered at the ground camera and then learns to select features along the height dimension, pooling this 3D point cloud into a point plane. This enables us to trace which features in the ground-level image contribute to the point plane. Next, our method computes the correspondences between the generated points from the ground view and points sampled from the aerial view and solves the relative pose between the two point planes using Procrustes alignment [30] with sparse correspondences. Notably, without any ground truth local feature correspondences between ground and aerial images, our method learns reasonable matches in a weakly supervised manner using only the ground truth camera pose.

Concretely, our main contributions are[1]: (i) A novel fine-grained cross-view localization method that matches local features between ground and aerial images. Experiments show that our method achieves superior localization accuracy, reducing the mean localization error by 28% on a challenging benchmark compared to the previous state-of-the-art. (ii) Our proposed feature selection along height enables the interpretability of the localization results. With supervision only on camera pose, our method learns to match local features between ground and aerial images in a weakly supervised manner. Qualitative results show that our matching is semantically consistent across ground and aerial views.

2. Related work

Deep cross-view localization methods typically start with a Siamese-like feature extractor. Built on top of the extracted features, methods have two different objectives for constructing representations to localize the ground images.

Representing what should be seen by a ground camera positioned at candidate locations in an aerial image is a common formulation [20, 46, 47]. These methods render descriptors at candidate locations in the aerial view and then match them with the descriptor of the ground view to generate a localization heat map. During this step, [20, 47] constructs orientation-aware descriptors to estimate the camera's orientation jointly. The idea of learning a global descriptor to summarize expected visual content is also used in other BEV map-based localization [16, 17, 25] and cross-view image retrieval [10, 18, 24, 31, 32, 45, 49]. Although the heat maps indicate localization uncertainty, interpreting which objects contribute to localization remains challenging due to the use of abstract global image descriptors.

Representing what is here refers to constructing a BEV representation from the ground view, where each BEV cell captures details of the local vicinity [8, 9, 29, 34, 36, 39, 42, 43]. Localization is then achieved in two ways. The first approach sweeps the ground view BEV representation across the aerial BEV representation and identifies the pose with the highest alignment [8, 9, 29, 34]. However, supervising element-wise alignment between two BEV representations encourages the model's ground branch to hallucinate information behind occlusions. The second approach estimates the relative pose between ground and aerial BEV features using dense [36, 43] or sparse [39, 42] matches between them. However, [36, 39] apply homography in BEV mapping, forcing the model to ignore above-ground objects. [42] use transformers to resolve ambiguity along height, but it assumes knowledge of on-ground pixels in the ground view. In practice, the height of horizon in the image depends on occlusions along the depth, so this assumption does not hold. Moreover, [36, 43] use learned modules to predict the pose, making the results hard to interpret, as it is unclear how specific features contribute to the prediction.

Ground-level image relative pose estimation is a simpler yet closely related task. The classic pipeline [14] involves detecting and describing key points in two images, matching these 2D key points, and estimating the 6 DoF pose based on the matches within RANSAC loops [11]. Recent advancements [3, 21, 41] lift the two images to two sets of 3D points and align the two 3D point clouds for relative pose estimation. For ground-to-aerial cross-view localization (which focuses on 3 DoF pose), one can generate two sets of points on a BEV plane and then align two point planes based on their sparse point-wise matches.

3. Methodology

Given a ground-level image G and an aerial image A that covers the ground camera's local surroundings, the objective of fine-grained cross-view localization is to estimate the 3 DoF pose $\boldsymbol{p} = [\boldsymbol{t}, o]$ of the ground camera, where $\boldsymbol{t}$ represents the pixel coordinates of camera's location in the aerial image A, and o denotes the camera's yaw orientation.

3.1. Overview

Motivation: Our goal is to achieve fine-grained cross-view localization by matching fine-grained features across views. However, this matching has two main challenges: (i) Each pixel in the aerial image corresponds to a vertical ray in 3D (assuming parallel projection). The matching method needs to identify the height along this vertical line observed in the ground view that matches aerial view content. However, existing datasets do not have such ground truth correspondences; (ii) Matchable information is sparse since not all content in the aerial view is visible from the ground view due to occlusions. Besides, dynamic objects are unmatchable because the images are not captured simultaneously.

[1]Code is available at: https://github.com/vita-epfl/FG2

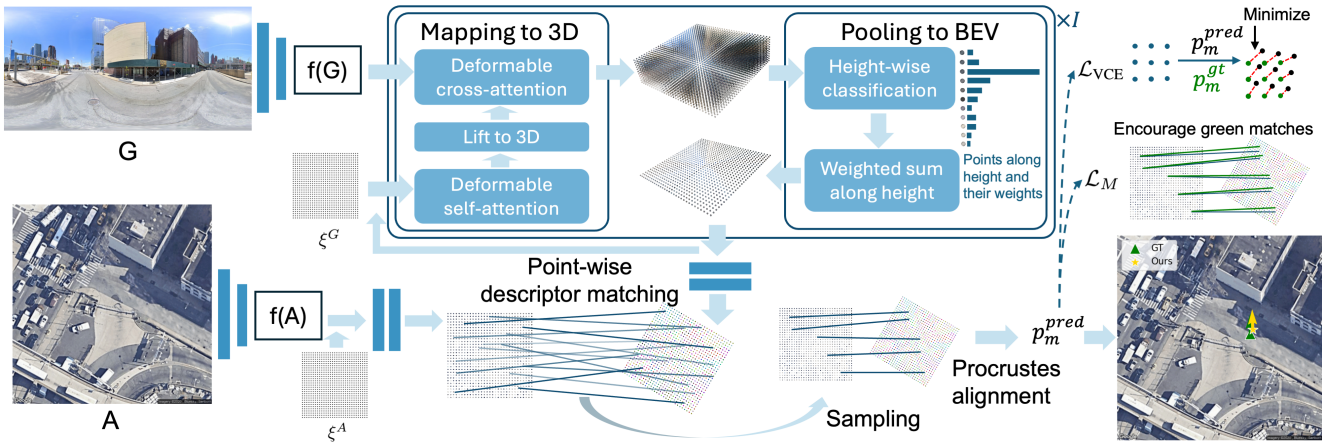


Figure 2. Overview of our proposed method and the objectives of the loss functions used. We define two sets of points, ξ^G and ξ^A, on a BEV plane, and our method generates a descriptor for each point. For the ground view, this involves mapping the image feature $f(G)$ to 3D and then selecting the important features along height. Next, we compute pairwise matching scores between the two point sets. The pose is then computed using Procrustes alignment based on the sampled matches. $\mathcal{L}_{\text{VCE}}$ minimizes the difference between a virtual point set transformed using the predicted pose and the ground truth pose. $\mathcal{L}_M$ encourages correspondences found using the ground truth pose.

We address the above challenges with a model that is end-to-end trained using only the ground truth camera pose. To resolve the height ambiguity, our method first lifts the ground image to a 3D point cloud and then selects the 3D points along the vertical direction whose feature best aligns with the aerial view. This selection is learned from data such that the selected feature should produce a matching result in the next step that aligns with the ground truth camera pose. This selection pools the 3D point cloud into a 2D point plane. We then match these 2D points with 2D points sampled from the aerial view. Given that the matchable information is sparse, we select only the top correspondences to compute the ground camera pose. Notably, our explicit selection of the vertical points allows us to trace which objects in the ground image are matched to those in the aerial image, providing interpretable localization results.

Formulation: We estimate the ground camera pose $\boldsymbol{p} = [\boldsymbol{t}, o]$ by aligning two point planes, i.e., computing their metric relative pose $\boldsymbol{p}_m = [\boldsymbol{t}_m, o]$. One point plane has its origin at the ground camera and the other has its origin at the center of the aerial image. Given the known Ground Sampling Distance (GSD) in meters per pixel of the aerial image, we compute $\boldsymbol{t}$ as $\boldsymbol{t} = \boldsymbol{t}_m/\text{GSD} + \boldsymbol{s}$, where $\boldsymbol{s}$ is the offset between the center and the top-left of the aerial image.

As shown in Fig. 2, our method first extracts ground and aerial image features $f(G)$ and $f(A)$. Similar to recent advances in ground-level image matching [3, 6], we adopt pre-trained DINOv2 features[2] [27] as $f(G)$ and $f(A)$. To construct the ground and aerial point planes, we define two sets of N points, $\xi^G = \{(x_1^G, y_1^G), ..., (x_N^G, y_N^G)\}$ and $\xi^A = \{(x_1^A, y_1^A), ..., (x_N^A, y_N^A)\}$, regularly spanning an area $[-L/2, L/2]^2$, and L is the side length of the aerial image in meters. (x, y) denotes the coordinates of each point, and the origins of ξ^G and ξ^A are located at the ground camera and the center of the aerial image, respectively. Next, there are two main steps to infer the 3 DoF relative pose $\boldsymbol{p}_m$ between ξ^G and ξ^A: First, we generate a descriptor for each point in ξ^G and ξ^A (Sec. 3.2). Second, we compute $\boldsymbol{p}_m$ based on the point matching scores (Sec. 3.3).

[2]Our Appendix will show that directly matching ground and aerial DINO features without our designed BEV points matching does not work.

3.2. BEV mapping

Given the two point sets, ξ^G and ξ^A, and ground and aerial image features, $f(G)$ and $f(A)$, our BEV mapping generates for each ground and aerial point, (x_n^G, y_n^G) and (x_n^A, y_n^A), a descriptor, d_n^G and d_n^A, where $n \in [1, ..., N]$. For the ground view, we first map the ground image feature to 3D and then pool the 3D feature to BEV.

Mapping to 3D: We follow an approach similar to recent 3D object detection [22, 40] to map image features to 3D. Each 2D point in ξ^G is lifted to a pillar of M 3D points spanning a given height range. These 3D points are then projected to the ground image to retrieve features.

Specifically, we use deformable attention [52]. For the n-th point, (x_n^G, y_n^G), in ξ^G, we first associate it with a learnable query q_n^G. Next, we find for the m-th 3D point $\mathcal{X}_{n,m}$, $m \in [1, ..., M]$, its projected pixel coordinates $\boldsymbol{r} = (u_{n,m}, v_{n,m})$ in $f(G)$. Then, $\mathcal{X}_{n,m}$ retrieves features from $f(G)$ using deformable cross-attention,

$$F_{\mathcal{X}_{n,m}} = \sum_{j=1}^{J} W_j \sum_{k=1}^{K} A_{j,k} \times f(G)_{\boldsymbol{r}+\Delta\boldsymbol{r}_{j,k}}. \quad (1)$$

In Eq. 1, $F_{\mathcal{X}_{n,m}}$ is the feature for the 3D point $\mathcal{X}_{n,m}$. The deformable attention has J attention heads with attention weights W_j generated from the query q_n^G. Within each head, there are K offsets $\Delta \boldsymbol{r}_{j,k}$ and their weights $A_{j,k}$ mapped from the query q_n^G, where $\sum_{k=1}^{K} A_{j,k} = 1$. The term $f(G)_{\boldsymbol{r}+\Delta \boldsymbol{r}_{j,k}}$ denotes the bilinearly interpolated feature at the location $\boldsymbol{r} + \Delta \boldsymbol{r}_{j,k}$ in $f(G)$. These learnable offsets $\Delta \boldsymbol{r}_{j,k}$ allow the model to capture informative features around the projected pixel coordinates $\boldsymbol{r}$.

We retrieve features for all 3D points. If the ground image G is panoramic, all points can find their 2D pixels inside $f(G)$. When G has a limited field of view, we record any 3D points projected outside $f(G)$ and discard them when performing ground-aerial point matching in later pose estimation. Similar to [22], before mapping to 3D, the learnable query q_n^G first undergoes a deformable self-attention layer to gather information from neighboring 2D points, i.e., using Eq. 1 by replacing $f(G)$ with the queries of 2D points.

Pooling to BEV: Next, we pool the 3D points to the BEV plane and generate a descriptor d_n^G for each 2D point (x_n^G, y_n^G). Typically, this is done by summing [22] or taking max [29] feature among the M 3D points over height. However, we argue that both approaches discard information about which height contributes to the BEV feature, as the dominant feature in summation or the argmax can vary across channels. To identify the ground feature along height that contributes to matching with the aerial view, we propose to learn the feature selection along height. Our ablation study will show it also improves localization accuracy.

Given the 3D points $\mathcal{X}_{n,m}$ and their features $F_{\mathcal{X}_{n,m}}$, we treat the BEV pooling problem as a classification problem. We classify the importance of the M 3D points along the height dimension and the feature g_n^G for the 2D point (x_n^G, y_n^G) is then a weighted sum of the M features,

$$g_n^G = \sum_{m=1}^{M} \frac{e^{l(F_{\mathcal{X}_{n,m}})}}{\sum_{m=1}^{M} e^{l(F_{\mathcal{X}_{n,m}})}} F_{\mathcal{X}_{n,m}}, \tag{2}$$

where l maps each $F_{\mathcal{X}_{n,m}}$ to a single value. We highlight again that we do not have ground truth for this classification, and the model needs to learn the feature selection backpropagated from the supervision on the camera pose.

In practice, we treat the mapping to 3D and pooling to BEV as a unified module and iterate it I times (using g_n^G as the new query q_n^G), similar to [22]. Finally, we obtain the descriptor d_n^G of the 2D point (x_n^G, y_n^G) by applying a projection head on its final feature, $d_n^G = \text{proj}(g_n^G)$, and the projection head consists of ResNet blocks [15], a small self-attention layer, and a L_2-normalization layer.

Aerial BEV sampling: Since the aerial view already provides a BEV perspective, assigning features to points in ξ^A only requires looking up the feature at the corresponding locations in the extracted aerial image feature map $f(A)$. Specifically, for each point (x_n^A, y_n^A), its feature g_n^A is a bilinear interpolation at location $(\frac{(x_n^A, y_n^A)}{\text{GSD}} + \boldsymbol{s})$ in $f(A)$. Similar to the ground branch, we use a projection head with the same architecture to map g_n^A to a descriptor d_n^A.

3.3. Pose estimation

We compute the metric pose $\boldsymbol{p}_m$ by first matching descriptors of two point sets ξ^G and ξ^A, and then use Procrustes alignment [30] based on matched correspondences.

Point-wise descriptor matching: Given the ground and aerial point descriptors, $\{d_1^G, ..., d_N^G\}$ and $\{d_1^A, ..., d_N^A\}$, we first compute their pair-wise similarly score c_{n_G, n_A},

$$c_{n_G, n_A} = \frac{\text{cosine similarity}(d_{n_G}^G, d_{n_A}^A)}{\tau}, \tag{3}$$

where τ is a temperature parameter and $n_G \in [1, N], n_A \in [1, N]$ are indexes for ground and aerial descriptors.

Since the descriptor matching probability is the probability that they are mutual nearest neighbors, we use the dual-Softmax operator as in [3, 6, 37, 38] to compute the product of the probability of $d_{n_G}^G$ matched to $d_{n_A}^A$ for a fixed n_G and the probability of $d_{n_A}^A$ matched to $d_{n_G}^G$ for a fixed n_A,

$$D_{n_G, n_A} = \frac{e^{c_{n_G, n_A}}}{\sum_{n_A=1}^{N} e^{c_{n_G, n_A}} + e^b} \times \frac{e^{c_{n_G, n_A}}}{\sum_{n_G=1}^{N} e^{c_{n_G, n_A}} + e^b}. \tag{4}$$

The term D_{n_G, n_A} denotes the probability of the n_G-th ground point $(x_{n_G}^G, y_{n_G}^G)$ and the n_A-th aerial point $(x_{n_A}^A, y_{n_A}^A)$ are matched and b is a learnable dustbin parameter to allow unmatched points to be assigned to it [28].

Pose estimation by Procrustes alignment: We use Kabsch solver [19] (details in Appendix) to compute the 3 DoF pose $\boldsymbol{p}_m$ between the two point sets, ξ^G and ξ^A, given their correspondences. This process is differentiable [2, 5].

Since not all information in the ground and aerial views are matchable, we sample a subset of N_S correspondences for Procrustes based on the descriptor matching probability D. Our ablation study will show that sampling a subset provides better localization accuracy than using all points.

Inference with RANSAC: As our pose estimation requires sampling from the matching probability D, we can incorporate robust estimators, such as RANSAC [11], in inference. We sample the descriptor matching probability R times. In each iteration r, we count the inliers for the pose $\boldsymbol{p}_m^r$, computed using the sampled correspondences. A ground-aerial point correspondence $(x_{n_G}^G, y_{n_G}^G) \leftrightarrow (x_{n_A}^A, y_{n_A}^A)$ is considered as an inlier if,

$$\sqrt{(x_{\tilde{n_G}}^A - x_{n_A}^A)^2 + (y_{\tilde{n_G}}^A - y_{n_A}^A)^2} < \text{threshold}, \tag{5}$$

where $(x_{\tilde{n_G}}^A, y_{\tilde{n_G}}^A)$ is acquired by transforming the ground point $(x_{n_G}^G, y_{n_G}^G)$ using $\boldsymbol{p}_m^r$. Finally, we find the pose with the most inliers and use all inliers to compute $\boldsymbol{p}_m^{pred}$.

3.4. Loss

Our loss combines a Virtual Correspondence Error loss adapted from [1] and a matching loss using a weight β,

$$\mathcal{L} = \mathcal{L}_{\text{VCE}} + \beta \mathcal{L}_M. \tag{6}$$

Our $\mathcal{L}_{\text{VCE}}$ provides supervision signal on the predicted pose $\boldsymbol{p}_m$. As shown in Fig. 2 right, we define a set of 2D points (in blue) spanning a BEV space $[-L_V/2, L_V/2]^2$, where L_V is a hyperparameter. Then, we use the ground truth $\boldsymbol{p}_m^{gt}$ and predicted pose $\boldsymbol{p}_m^{pred}$ to find the true (in green) and predicted (in black) correspondences for the blue points. Our $\mathcal{L}_{\text{VCE}}$ then minimizes the mean Euclidean distance between all corresponding green and black points.

Besides supervising the camera pose, we use $\mathcal{L}_M$ to provide a direct supervision signal on the ground-aerial point descriptor matching. As the matchable information between ground and aerial images is sparse, not all points in ξ^G should find a correspondence in ξ^A and vice versa. Therefore, we take the N_S sampled correspondences to compute $\mathcal{L}_M$. For each estimated correspondence $(x^G_{n_G}, y^G_{n_G}) \leftrightarrow (x^A_{n_A}, y^A_{n_A})$, we find for the ground point $(x^G_{n_G}, y^G_{n_G})$ the ground truth aerial point $(x^A_{\hat{n_G}}, y^A_{\hat{n_G}})$ using $\boldsymbol{p}_m^{gt}$ and then compute an infoNCE loss [26] for all N_S correspondences,

$$\mathcal{L}^G_{\text{infoNCE}} = -\log \frac{\sum_{n_G=1}^{N_S} e^{c_{n_G,\hat{n_G}}}}{\sum_{n_G=1}^{N_S}\sum_{n_A=1}^{N} e^{c_{n_G,n_A}}}, \tag{7}$$

where c is the descriptor similarity score (Eq. 3). Similarly, we compute an infoNCE loss by finding the ground truth ground point $(x^G_{\hat{n_A}}, y^G_{\hat{n_A}})$ for each aerial point $(x^A_{n_A}, y^A_{n_A})$,

$$\mathcal{L}^A_{\text{infoNCE}} = -\log \frac{\sum_{n_A=1}^{N_S} e^{c_{n_A,\hat{n_A}}}}{\sum_{n_A=1}^{N_S}\sum_{n_G=1}^{N} e^{c_{n_G,n_A}}}. \tag{8}$$

Finally, $\mathcal{L}_M$ is an average of the two infoNCE losses,

$$\mathcal{L}_M = (\mathcal{L}^G_{\text{infoNCE}} + \mathcal{L}^A_{\text{infoNCE}})/2. \tag{9}$$

4. Experiments

We first introduce the used datasets and evaluation metrics. Then, our implementation details are presented. Next, we compare our method to previous state-of-the-art and show ground-aerial fine-grained feature matching results. Finally, we conduct an ablation study on key hyperparameters.

4.1. Datasets and evaluation metrics

We use two datasets in our experiments.

VIGOR [51] contains ground-level panoramas and aerial images collected in four cities in the US. It has two settings: same-area and cross-area. In the same-area setting, the training and test images come from all cities, whereas, in the cross-area setting, training images are from two cities, and test images are from the other two. Consistent with previous evaluation protocol [20, 47]: we include both known and unknown orientation settings; use the positive samples, i.e., ground images located in the center $1/4$ region in the aerial images; split 20% of training data for validation and ablation study; and use the ground truth labels from [20].

KITTI [12] provides front-facing images with a limited field of view collected in Germany. [33] provides aerial images for KITTI. Similar to VIGOR [51], the data is split into same-area and cross-area. We adopt the common setting [31] that ground images are located in the center 40 m $\times$ 40 m area of their corresponding aerial images, and there is an orientation prior with noise between $\pm 10°$.

Metrics: We report mean and median localization and orientation errors. Following [43, 47], on KITTI, we further decompose localization errors into longitudinal and lateral components based on the driving direction, and report the percentage of samples within given error thresholds.

4.2. Implementation details

On VIGOR, we have 41×41 points in both ξ^G and ξ^A, and each ground 2D point is lifted to 11 3D points, regularly distributed from -20 m to 20 m (assuming the camera is at 0 m). On KITTI, we use the same number of points in ξ^A. Since KITTI's ground image has a limited field of view, we create ground points only in front of the camera, resulting in 21×41 points in ξ^G. The 11 3D points span from -25 m to 25 m to ensure all content in the image can be mapped to 3D. Similar to [22], we iterative the unified BEV mapping module $I = 6$ times. As in [3], we use $\tau = 10$ in Eq. 3. We sample $N_S = 1024$ correspondences and keep their matching scores as weights when solving Procrustes alignment.

On both datasets, we use AdamW [23] with a learning rate of 1×10^{-4}, and our batch size is 24. The weight β in our loss is set to 1, and we use 10×10 2D points over $[-2.5\text{ m}, 2.5\text{ m}]^2$ space ($L_V = 5$ m) when computing $\mathcal{L}_{\text{VCE}}$.

4.3. Quantitative results

VIGOR: We compare our method against previous state-of-the-art, including both global descriptor-based [20, 47] and local feature-based [34, 36, 43] methods.

As shown in Tab. 1, our method achieves a considerably lower mean localization error than others when the orientation is known. Compared to the previous state-of-the-art, HC-Net [43], our method reduces the mean localization error by 26% and 28% on the same-area and cross-area test sets, respectively. Our median errors are on par with those of previous best-performing methods, DenseFlow [36] and GGCVT [34], while having substantially lower mean errors.

Unknown orientation poses a greater challenge. GGCVT and HC-Net did not report results for this setting. The

Orien.	Methods	Same-area				Cross-area			
		↓ Localization (m)		↓ Orientation (°)		↓ Localization (m)		↓ Orientation (°)	
		Mean	Median	Mean	Median	Mean	Median	Mean	Median
Known	SliceMatch [20]	5.18	2.58	-	-	5.53	2.55	-	-
	CCVPE [47]	3.60	1.36	-	-	4.97	1.68	-	-
	GGCVT [34]	4.12	1.34	-	-	5.16	1.40	-	-
	DenseFlow [36]	3.03	**0.97**	-	-	5.01	2.42	-	-
	HC-Net [43]	2.65	1.17	-	-	3.35	1.59	-	-
	Ours	**1.95**	1.08	-	-	**2.41**	**1.37**	-	-
Unknown	SliceMatch [20]	6.49	3.13	25.46	4.71	7.22	3.31	25.97	4.51
	CCVPE [47]	**3.74**	**1.42**	12.83	6.62	**5.41**	**1.89**	27.78	13.58
	DenseFlow [36]	4.97	1.90	**11.20**	1.59	7.67	3.67	**17.63**	2.94
	Ours	8.95	7.32	15.02	2.94	10.02	8.14	31.41	5.45
	Ours†	3.78	1.70	12.63	**1.44**	5.95	2.40	28.41	**2.20**

Table 1. VIGOR test results. Ours† means using our method with a two-step inference to boost the performance. **Best in bold.**

global descriptor-based method CCVPE [47] outperforms all local feature-based methods, including ours. CCVPE has a localization branch and an orientation branch. Its localization branch is designed to be invariant to different orientations. While our localization and orientation are computed jointly. When orientation is unknown, our method must distinguish a tree along one viewing direction from all other trees in the scene, posing a challenging learning task. In contrast, when orientation is known, our model can focus on distinguishing objects along specific rays.

In practice, unknown orientation is less common, as we can obtain orientation estimates from a phone's built-in compass or temporal information. Nevertheless, we emphasize that, since our method accurately estimates orientation, we can use the model's prediction to rotate the ground panorama to a roughly orientation-aligned setting. We then apply another model, trained with a small orientation noise range, e.g., $\pm 20°$, to estimate the pose again. This simple two-step inference largely reduces the gap to state-of-the-art when testing with unknown orientation.

KITTI: Our method outperforms the previous state-of-the-art w.r.t. mean localization errors, while having slightly higher median errors, see Tab. 2. Compared to HC-Net [43], our method achieves comparable performance when testing on images from the same area but shows much lower mean and median errors when testing on images from different areas, demonstrating better generalization capability.

Since KITTI images have a limited field of view, localization in the longitudinal direction is more challenging than that in the lateral direction. Consequently, all methods have high recall in the lateral direction, and ours achieves the highest. For orientation prediction, our method performs slightly worse than the baselines. Most KITTI images view along the direction of the road, which creates a bias that other methods can easily learn. As the full aerial feature directly contributes to their output, simply predicting the road orientation leads to very low orientation errors. In contrast, our method must infer the pose from sparse matching results, making it more challenging to exploit this bias.

4.4. Qualitative results

Fine-grained feature matching: A major advantage of our method is its interpretability. Global descriptor-based methods [20, 47] condense the entire ground image into a single descriptor, making it unclear which ground local feature matches with which aerial local feature. Previous local feature-based methods [36, 43] match features in BEV space, but they do not explicitly track which ground feature along the height contributes to the BEV representation. More importantly, they use a learned module to output the camera pose based on the matching scores. We observed with HC-Net [43] that if we swap the order of ground and aerial features, its pose estimator does not produce an inverse pose of the unswapped version. This makes it unclear how feature matching influences the localization result.

In contrast, our estimated pose is a direct result of the sampled matching scores. Meanwhile, our proposed feature selection along height allows us to track which ground features in the image contribute to the matching in BEV space, making localization more interpretable.

We visualize the top 20 matched ground-aerial correspondences in Fig. 3. Overall, the matched content is semantically consistent. In (a) and (b), the points around zebra crossings in the ground view are matched to the zebra crossings in the aerial view. In (c) and (d), the points near the curbs and road markings in the ground view are matched to those curbs in the aerial view. Given that ground and aerial images capture the scene from perpendicular perspectives, our model matches building facades in the ground image to roofs in the aerial view, see Fig. 3 (b), (c), (e) and (f), similar

	KITTI	↓ Loc. (m)		↑ Lateral (%)		↑ Long. (%)		↓ Orien. (°)		↑ Orien. (%)	
		Mean	Median	R@1m	R@5m	R@1m	R@5m	Mean	Median	R@1°	R@5°
Same-area	GGCVT [34]	-	-	76.44	98.89	23.54	62.18	-	-	**99.10**	**100.00**
	CCVPE [47]	1.22	0.62	97.35	99.71	77.13	97.16	0.67	0.54	77.39	99.95
	HC-Net [43]	0.80	0.50	99.01	99.73	**92.20**	**99.25**	**0.45**	0.33	91.35	99.84
	DenseFlow [36]	1.48	**0.47**	95.47	99.79	87.89	94.78	0.49	**0.30**	89.40	99.31
	Ours	**0.75**	0.52	**99.73**	**100.00**	86.99	98.75	1.28	0.74	61.17	95.65
Cross-area	GGCVT [34]	-	-	57.72	91.16	14.15	45.00	-	-	**98.98**	**100.00**
	CCVPE [47]	9.16	**3.33**	44.06	90.23	23.08	64.31	**1.55**	**0.84**	57.72	96.19
	HC-Net [43]	8.47	4.57	75.00	97.76	**58.93**	**76.46**	3.22	1.63	33.58	83.78
	DenseFlow [36]	7.97	3.52	54.19	91.74	23.10	61.75	2.17	1.21	43.44	89.31
	Ours	**7.45**	4.03	**89.46**	**99.80**	12.42	55.73	3.33	1.88	30.34	81.17

Table 2. KITTI test results. We did not include the almost pixel-perfect (KITTI aerial image's GSD: 1 pixel = 0.2 m) localization result from [42], as we cannot reproduce it because of the unavailability of the code. **Best in bold.**

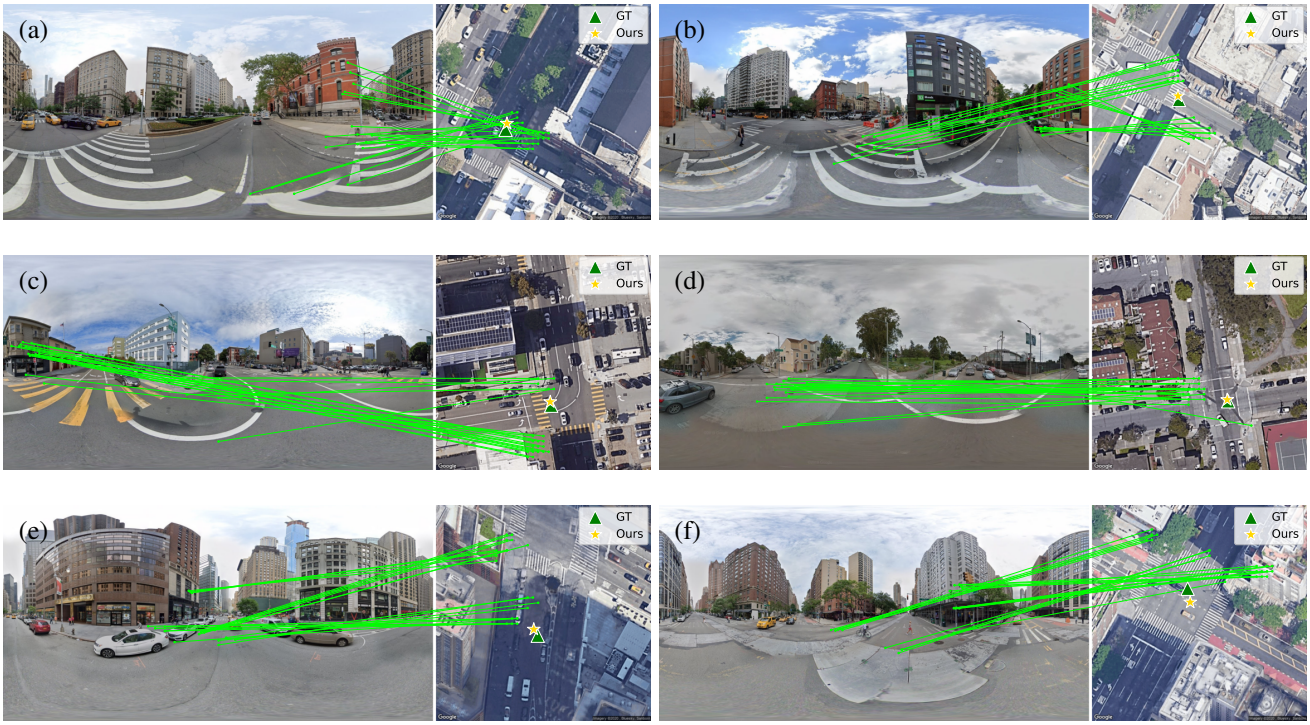


Figure 3. Fine-grained feature matching results on VIGOR with known orientation. We show the 20 matches with the highest similarity scores. We find the 3D points using the selected height in the last pooling to BEV step and then project those points to the ground image.

to how humans would do when localizing themselves. We emphasize again that this matching is learned in a weakly supervised manner, as we have only the ground truth camera pose and no explicit local feature correspondences.

Visualization of selected height: Our fine-grained feature matching is enabled by the proposed feature selection along height. Next, we visualize the selected height when pooling the ground 3D points onto the BEV plane in Fig. 4.

The first panorama shows a road in a roughly northeast-southwest direction, with buildings on the side. The height map shows that the model focuses on lower-height features along the road while paying attention to higher points at the sides of the road. Similarly, in the second example, the road runs from northwest to southeast, and our model again focuses on on-ground features along the road and shifts focus to higher points where there are buildings. By supervising only the camera pose, our model learns to select appropriate features along the height based on the image content.

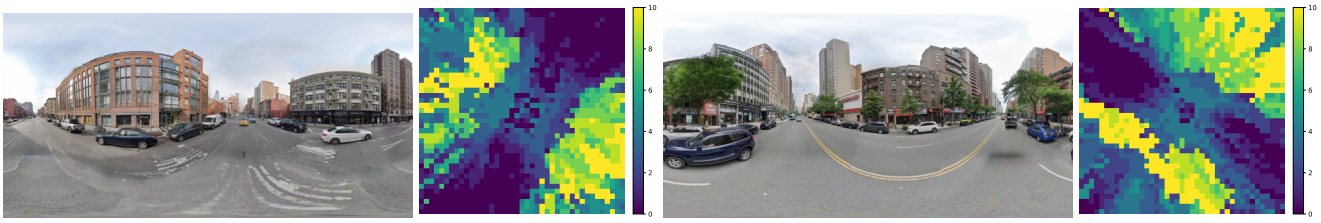


Figure 4. Visualization of selected height indexes when pooling the ground 3D points to BEV. Each cell in the BEV map corresponds to a pillar of 3D points, and the color denotes the selected height (larger values mean higher points). The ground camera is located at the center of the BEV height map. The center vertical line in the panorama corresponds to a line originating from the center of the BEV to the top.

4.5. Ablation study

Next, an extensive ablation study is conducted on the VIGOR same-area validation set with known orientation to investigate our key hyperparameters: the number of ground and aerial points N, the weight β on our matching loss $\mathcal{L}_M$, the number of sampled correspondences N_S for Procrustes alignment, and the effect of replacing our proposed selection along height with summing or taking the max feature.

As expected, our method benefits from a larger number of ground and aerial points N. We observe an increase in localization performance when increasing N from 21×21 to 41×41, and the performance saturates after 41×41, see Tab. 3. Thus, we use $N = 41 \times 41$. When excluding our matching loss $\mathcal{L}_M$ by setting β to 0, we observe a performance drop compared to our default setting. Besides, the model converges more slowly without direct supervision on descriptor matching. When β is too large, i.e., $\beta = 10$, the supervision signal on the predicted pose becomes less dominant, resulting in lower localization accuracy.

In [9, 34], the pose is inferred based on the dense matching result of ground and aerial features. Similarly, we tested using all point correspondences, incorporating their matching scores as weights for Procrustes alignment. However, this leads to higher errors, see "No sampling" in Tab. 3. Since matchable information between ground and aerial images is sparse, not all correspondences should contribute to pose estimation. Although unmatched correspondences may receive lower weights, a large number of such correspondences can still negatively impact accuracy. We find our localization accuracy is not significantly influenced by N_S when testing $N_S = 64, 128, 256, 512, 1024$, except that the model with $N_S = 64$ performs slightly worse. However, a larger N_S results in a smoother training loss curve.

Instead of using our proposed feature selection when pooling the 3D points to BEV, we also tried simply summing [22] or taking the maximum [29] feature over the 3D points along the height. As shown in the last two rows of Tab. 3, both formulations yield worse localization accuracy. Moreover, since the sum and max operations lose track of which features contribute to the BEV, they reduce the interpretability of the model.

Hyperparameters		↓ Localization (m)	
N	Settings	Mean	Median
21×21	default	2.50	1.48
31×31	default	2.26	1.25
41×41	default	**2.17**	1.18
51×51	default	2.19	**1.16**
41×41	$\beta = 0$	2.63	1.45
41×41	$\beta = 10$	2.54	1.38
41×41	No sampling	2.55	1.45
41×41	sum over height	2.34	1.27
41×41	max over height	2.24	1.25

Table 3. Ablation study. Default: $\beta = 1$, sampling with $N_S = 1024$, and learning feature selection over height. **Best in bold.**

5. Conclusion

We propose a novel fine-grained cross-view localization method that matches fine-grained features across views. Our model learns semantically consistent matches in a weakly supervised manner, requiring only the camera pose for training. Our method estimates the ground camera's pose by aligning a generated ground point plane with a sampled aerial point plane using Procrustes alignment. This process involves mapping ground image features to a 3D point cloud and then pooling the 3D points into a BEV plane. Instead of the standard approach of summing or taking the maximum feature along the height dimension when pooling to BEV, we proposed to learn the feature selection. This approach allows us to trace which ground image features contribute to the BEV representation and also improves localization accuracy. Experimental results show that our method surpasses previous state-of-the-art methods by a large margin, reducing the mean localization error by 28% on the challenging VIGOR cross-area test set. When orientation is unknown, we propose a two-step inference: first, using one model to estimate the orientation and bring the input to a roughly orientation-aligned setting; then, using another model trained with small orientation noise for accurate pose estimation. This approach achieves performance comparable to previous state-of-the-art methods.

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. 5

[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. 4

[3] Axel Barroso-Laguna, Sowmya Munukutla, Victor Adrian Prisacariu, and Eric Brachmann. Matching 2d images in 3d: Metric relative pose from metric correspondences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4852–4863, 2024. 2, 3, 4, 5

[4] Boaz Ben-Moshe, Elazar Elkin, et al. Improving accuracy of GNSS devices in urban canyons. In *Proceedings of the 23rd Annual Canadian Conference on Computational Geometry*, pages 511–515, 2011. 1

[5] 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. 4

[6] Johan Edstedt, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Dedode: Detect, don't describe—describe, don't detect for local feature matching. In *International Conference on 3D Vision*, pages 148–157. IEEE, 2024. 3, 4

[7] Russell A Epstein and Lindsay K Vass. Neural systems for landmark-based wayfinding in humans. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1635):20120533, 2014. 1

[8] Florian Fervers, Sebastian Bullinger, Christoph Bodensteiner, Michael Arens, and Rainer Stiefelhagen. C-bev: Contrastive bird's eye view training for cross-view image retrieval and 3-dof pose estimation. *arXiv preprint arXiv:2312.08060*, 2023. 2

[9] 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. 1, 2, 8

[10] Florian Fervers, Sebastian Bullinger, Christoph Bodensteiner, Michael Arens, and Rainer Stiefelhagen. Statewide visual geolocalization in the wild. In *European Conference on Computer Vision*, pages 438–455. Springer, 2024. 2

[11] Martin A Fischler and Robert C Bolles. 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. 2, 4

[12] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research*, 2013. 5

[13] Google. Google maps static api, 2024. 1

[14] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 2

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 4

[16] Henry Howard-Jenkins and Victor Adrian Prisacariu. Lalaloc++: Global floor plan comprehension for layout localisation in unvisited environments. In *European Conference on Computer Vision*, pages 693–709. Springer, 2022. 2

[17] Henry Howard-Jenkins, Jose-Raul Ruiz-Sarmiento, and Victor Adrian Prisacariu. Lalaloc: Latent layout localisation in dynamic, unvisited environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10107–10116, 2021. 2

[18] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 7258–7267, 2018. 2

[19] 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. 4

[20] 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. 1, 2, 5, 6

[21] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, 2024. 2

[22] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European Conference on Computer Vision*, pages 1–18. Springer, 2022. 3, 4, 5, 8

[23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 5

[24] Li Mi, Chang Xu, Javiera Castillo-Navarro, Syrielle Montariol, Wen Yang, Antoine Bosselut, and Devis Tuia. Congeo: Robust cross-view geo-localization across ground view variations. In *European Conference on Computer Vision*, 2024. 2

[25] Zhixiang Min, Naji Khosravan, Zachary Bessinger, Manjunath Narayana, Sing Bing Kang, Enrique Dunn, and Ivaylo Boyadzhiev. Laser: Latent space rendering for 2d visual localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11122–11131, 2022. 2

[26] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 5

[27] 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

[28] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4938–4947, 2020. 4

[29] Paul-Edouard Sarlin, Eduard Trulls, Marc Pollefeys, Jan Hosang, and Simon Lynen. Snap: Self-supervised neural maps for visual positioning and semantic understanding. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2, 4, 8

[30] Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966. 2, 4

[31] 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. 2, 5

[32] Yujiao Shi, Liu Liu, Xin Yu, and Hongdong Li. Spatial-aware feature aggregation for image based cross-view geo-localization. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[33] Yujiao Shi, Xin Yu, Liu Liu, Dylan Campbell, Piotr Koniusz, and Hongdong Li. Accurate 3-dof camera geo-localization via ground-to-satellite image matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):2682–2697, 2022. 1, 5

[34] 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. 1, 2, 5, 6, 7, 8

[35] Yujiao Shi, Hongdong Li, Akhil Perincherry, and Ankit Vora. Weakly-supervised camera localization by ground-to-satellite image registration. In *European Conference on Computer Vision*, pages 39–57. Springer, 2024.

[36] Zhenbo Song, Xianghui Ze, Jianfeng Lu, and Yujiao Shi. Learning dense flow field for highly-accurate cross-view camera localization. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2, 5, 6, 7

[37] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8922–8931, 2021. 4

[38] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. *Advances in Neural Information Processing Systems*, 33:14254–14265, 2020. 4

[39] Shan Wang, Yanhao Zhang, Akhil Perincherry, Ankit Vora, and Hongdong Li. View consistent purification for accurate cross-view localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8197–8206, 2023. 1, 2

[40] Shiming Wang, Holger Caesar, Liangliang Nan, and Julian FP Kooij. Unibev: Multi-modal 3d object detection with uniform bev encoders for robustness against missing sensor modalities. In *IEEE Intelligent Vehicles Symposium*, pages 2776–2783. IEEE, 2024. 3

[41] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 2

[42] Shan Wang, Chuong Nguyen, Jiawei Liu, Yanhao Zhang, Sundaram Muthu, Fahira Afzal Maken, Kaihao Zhang, and Hongdong Li. View from above: Orthogonal-view aware cross-view localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14843–14852, 2024. 1, 2, 7

[43] Xiaolong Wang, Runsen Xu, Zhuofan Cui, Zeyu Wan, and Yu Zhang. Fine-grained cross-view geo-localization using a correlation-aware homography estimator. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2, 5, 6, 7

[44] Thomas Wolbers and Mary Hegarty. What determines our navigational abilities? *Trends in cognitive sciences*, 14(3): 138–146, 2010. 1

[45] Zimin Xia, Olaf Booij, Marco Manfredi, and Julian FP Kooij. Geographically local representation learning with a spatial prior for visual localization. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 557–573. Springer, 2020. 2

[46] 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. 1, 2

[47] Zimin Xia, Olaf Booij, and Julian FP Kooij. Convolutional cross-view pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 1, 2, 5, 6, 7

[48] Zimin Xia, Yujiao Shi, Hongdong Li, and Julian FP Kooij. Adapting fine-grained cross-view localization to areas without fine ground truth. In *European Conference on Computer Vision*, pages 397–415. Springer, 2024. 1

[49] Hongji Yang, Xiufan Lu, and Yingying Zhu. Cross-view geo-localization with layer-to-layer transformer. *Advances in Neural Information Processing Systems*, 34:29009–29020, 2021. 2

[50] Andi Zang, Runsheng Xu, Goce Trajcevski, and Fan Zhou. Data issues in high-definition maps furniture–a survey. *ACM Transactions on Spatial Algorithms and Systems*, 10(1):1–37, 2024. 1

[51] 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. 5

[52] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021. 3