2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Satellite Image Based Cross-view Localization for Autonomous Vehicle

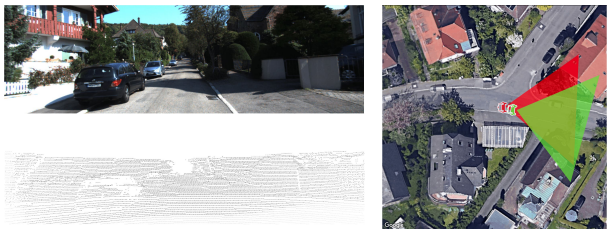
Shan Wang[1,2], Yanhao Zhang[1], Ankit Vora[3], Akhil Perincherry[3], and Hongdong Li[1]

***Abstract*— Existing spatial localization techniques for autonomous vehicles mostly use a pre-built 3D-HD map, often constructed using a survey-grade 3D mapping vehicle, which is not only expensive but also laborious. This paper shows that by using an off-the-shelf high-definition satellite image as a ready-to-use map, we are able to achieve cross-view vehicle localization up to a satisfactory accuracy, providing a cheaper and more practical way for localization. While the utilization of satellite imagery for cross-view localization is an established concept, the conventional methodology focuses primarily on image retrieval. This paper introduces a novel approach to cross-view localization that departs from the conventional image retrieval method. Specifically, our method develops (1) a Geometric-align Feature Extractor (GaFE) that leverages measured 3D points to bridge the geometric gap between ground and overhead views, (2) a Pose Aware Branch (PAB) adopting a triplet loss to encourage pose-aware feature extraction, and (3) a Recursive Pose Refine Branch (RPRB) using the Levenberg-Marquardt (LM) algorithm to align the initial pose towards the true vehicle pose iteratively. Our method is validated on KITTI and Ford Multi-AV Seasonal datasets as ground view and Google Maps as the satellite view. The results demonstrate the superiority of our method in cross-view localization with median spatial and angular errors within 1 meter and 1°, respectively.**

***Index Terms*— Cross-View localization, Pose Estimation, Deep Learning**

I. Introduction

Accurate vehicle localization plays an enabling role in autonomous driving. Although consumer-grade GPS devices have been widely used for vehicle localization, their performances degrade rapidly in GPS-compromised areas [1]. For instance, it is difficult to obtain reliable localization in urban areas with high-rise buildings. Other sensor modalities such as camera [2], [3], [4], and LiDAR [5], [6], [7], [8] are explored for achieving robust vehicle localization. Yet, the existing vehicle localization techniques critically rely on a pre-built 3D high-definition map. Both the acquisition and maintenance of such a 3D HD map are laborious and expensive, especially for rural areas where a mapping vehicle only visits rather infrequently. Cross-view Localization, by using off-the-shelf commercially-available satellite images as a map in spatial accordance with ground-view images captured by vehicle cameras, provides a cost-effective and promising solution. Recently, several works have been published in this front [9], [10], [11], [12], [13]. These cross-view localization methods, however, do not take full advantage of the satellite information. Instead, they only handle it via the conventional image-retrieval idea, hence they only achieve coarse localization.



(a) *Query* (b) *Reference*

Fig. 1: (a) Query information, including a ground-view image and corresponding 3D LiDAR points, and (b) reference satellite image. We aim to estimate the accurate 3-DoF pose of the ground-view camera. The initial and accurate 3-DoF poses are shown in (red) and (green), respectively.

Different from the image-retrieval-based cross-view methods, we propose a new fine-grained cross-view localization in this paper. Given a spatially-consistent satellite image, our method aims to estimate the accurate 3-DoF pose of the vehicle using a ground-view image (vehicle camera) and the 3D LiDAR points. Fig. 1 illustrates the setting of our method. To mitigate the domain gap between ground-view images and satellite images, we establish correspondences between the two views by projecting 3D LiDAR points onto their respective images. Given an initial coarse pose on the satellite map, we iteratively optimize the pose through feature matching.

Our Satellite Image Based Cross-view Localization (SIBCL) deep neural network consists of a Geometric-align Feature Extractor (GaFE) and two branches of objective functions, Pose-Aware Branch (PAB) and Recursive Pose Refine Branch (RPRB). More specifically, GaFE embeds ground-view and satellite images into the feature space with a shared weights encoder. It further establishes spatial-feature correspondences by projecting the 3D points onto the respective images. PAB employs a triplet loss [14] to differentiate the variation across point features (residual) between two views conditioned on the correct (ground truth) and incorrect (initial) pose. RPRB is tasked to iteratively optimize the initial pose towards the ground truth pose with the Levenberg–Marquardt (LM) algorithm. Moreover, a re-projection error is deployed on the optimized pose. It is noted that both objective branches supervise feature extraction but have a different focus. PAB encourages the correct pose estimation as well as penalizes for the incorrect. RPRB encourages the most correct (predict) pose close to the ground truth.

[1] Shan Wang, Yanhao Zhang, and Hongdong Li are with Australian National University. Shan.Wang@anu.edu.au.
[2] Shan Wang is also with Data61, CSIRO, Australia.
[3] Ankit Vora and Akhil Perincherry are with Ford Motor Company, Dearborn, USA.

In order to train and evaluate our method, we construct two cross-view localization datasets, KITTI-CVL and FordAV-CVL. These are composed of ground-view images from KITTI [15], Ford Multi-AV Seasonal [16] datasets respectively, and their spatial-consistent satellite counterparts from Google Map [17] according to image-wise GPS information. Extensive experiments on the proposed KITTI-CVL and FordAV-CVL datasets demonstrate that our SIBCL can accurately estimate vehicle positions with median errors being limited to within 1 meter in longitudinal/lateral shift and 1° in angular error.

The contributions of this paper are two-fold:

- a fine-grained cross-view localization method, SIBCL, that achieves accurate vehicle pose estimation with low spatial and angular errors.
- two branches of objectives design, RPRB encourages the predicted pose close to the ground truth, and PAB discriminates residuals across two views between the correct and incorrect pose.

II. Related work

Visual Localization. Intensive research has been done in the field of visual localization for autonomous driving. SLAM methodologies [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28] have traditionally been used for vehicle localization. They construct or update a geometrically accurate 3D map while simultaneously tracking the vehicle's location. The consecutive input sequence is essential since the geometrical accuracy relies on repeat observations of the same scene. As a result, SLAM methods are prone to error accumulation, which results in an estimation drift. In contrast, our method does not require a continuous series of images, and is able to estimate precise poses using a single query image paired with its corresponding 3D LiDAR points. Other localization algorithms leave the mapping problem to an existing 3D scene model [29], [30], [31], [32] or directly utilize pre-collected 3D HD maps [33]. Both the acquisition and maintenance of such a 3D model/map are laborious and expensive, especially for rural areas rarely visited by surveying vehicles.

Cross-view Visual Localization. Satellite imagery is widely available, well-maintained, and easy to access. Recent works [9], [10], [11], [12], [13], [34] resort to satellite images for localization. All these methods estimate correspondences based on image-level features, making the localization inside a satellite image impossible. As a result, they do not perform well in fine-grained localization. Miller et al. [35] proposed a cross-view SLAM method that uses semantically labeled LiDAR. Unlike our method, which focuses on localization-driven feature extraction, their approach relies heavily on semantic prior. Fervers et al. [36] proposed a cross-view localization method that ignores heading estimation and focuses on shift estimation. The most similar approach to ours is HighlyAccurate [37], a fine-grained cross-view localization method aimed at achieving 3-degree estimation. It projects dense features from the satellite map onto the ground-view, under the assumption that all pixels in ground-view images are located on the ground plane. The features of above-ground pixels are projected to the incorrect position due to this wrong assumption, limiting its overall performance. In contrast, for more accurate pose estimation, our method builds geometric correspondences across reliable 3D LiDAR points. Furthermore, we adopt an attention map to reduce the influence of dynamic objects.

III. Methodology

A. Overview

Given a coarse pose $\mathbf{P}_{init}$, our goal is to estimate the accurate 3-DoF pose $\mathbf{P}_{pre}$ of a ground-view camera using a ground-view image I^g and the 3D points in the ground-view (camera) domain $[x^g\ y^g\ z^g]^\top$. In our method, the 3D points are randomly sampled from valid LiDAR points. The framework of the proposed SIBCL is illustrated in Fig. 2. The GaFE employs a Convolutional Neural Network (CNN) to extract feature maps F^s and F^g from the satellite and ground-view images, respectively. We adopt a U-Net structure of CNN that aims to obtain feature maps with original resolution that benefits accurate pose estimation. We also compute spatial attention maps A^s, A^g to weight the feature maps, emphasizing pixels with potential correspondences between the two sets of images. Further, we project the 3D points onto the ground view images and satellite maps to obtain sparse pixel-level features and attention maps. Specifically, we obtain $F^g[p]$ and $A^g[p]$ from the ground-view images, and $F^s[p]_{\mathbf{P}}$ and $A^s[p]_{\mathbf{P}}$ from the satellite maps using the camera pose $\mathbf{P}$. Residual $r_{\mathbf{P}}$ and point weights $w_{\mathbf{P}}$ are calculated from these sparse representations across the two views. In PAB, a triplet loss is employed to narrow the residual by the ground truth pose $r[p]_{\mathbf{P}_{gt}}$ and widen that by the initial pose $r[p]_{\mathbf{P}_{init}}$. We only enable the PAB when the initial pose (incorrect pose) is significantly different from the ground truth pose. The RPRB is designed to iteratively optimize the predicted pose $\mathbf{P}_{pre}$ towards the ground truth pose $\mathbf{P}_{gt}$ using the LM algorithm.

B. Geometric-align Feature Extractor

The GaFE extracts a hierarchy of ground-view and satellite feature maps at multiple resolutions, $F^{g/s} = \{F_l^{g/s} \in \mathbb{R}^{h_l \times w_l \times c_l} | l = 1, \dots, L\}$ where l is the level of U-Net outputs, h_l, w_l, and c_l represent the height, width, and channel number of feature maps in each level. Each pixel-level feature representation among these feature maps are L_2 normalised in order to improve the robustness to, e.g., variance in illumination conditions and viewpoints. We further compute a spatial attention map $A^{g/s} = \{A_l^{g/s} \in \mathbb{R}^{h_l \times w_l}\} | l = 1, \dots, L\}$ by passing the un-normalized feature maps through a convolutional layer followed by a sigmoid activation function. This attention map is used to highlight pixels with potential cross-view correspondences. It assigns low score to temporal-inconsistent objects, like cars, and high scores to building edges and road marks that are identifiable from the satellite image. A visualization of the attention map is shown in Fig. 5.

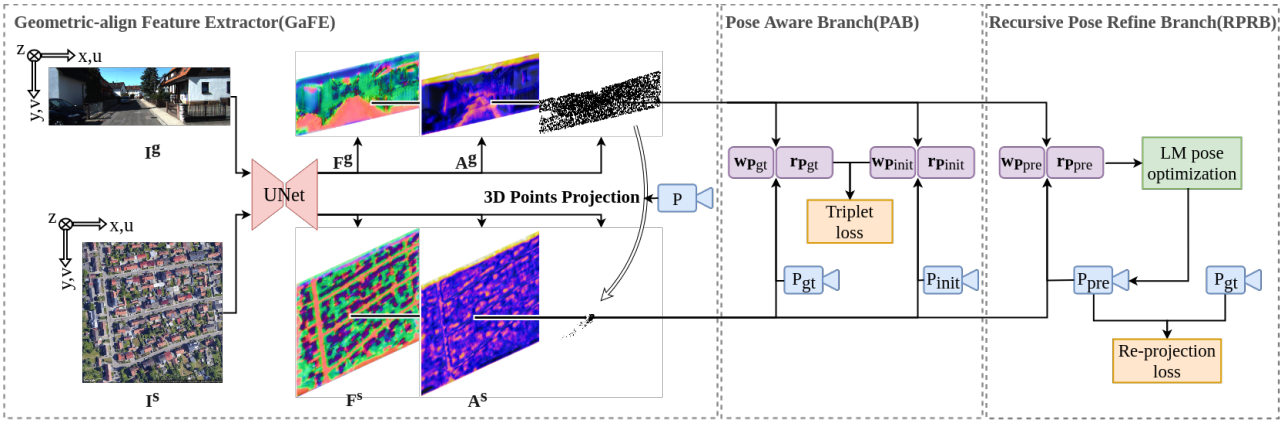


Fig. 2: Overview of SIBCL. GaFE extracts feature and attention maps from the ground and satellite views. Further, we obtain sparse pixel-level features and weights across the 3D points. The pose-aware features extraction is supervised in PAB by a triplet loss. In RPRB, the deep features are used to iteratively optimize the camera pose using the LM algorithm.

The coordinate systems of satellite and ground-view images are illustrated on the left of Fig. 2. The coordinates system of the satellite image is that the x^s-axis points to the east and the y^s-axis points to the south, and following the right-hand rule, the z^s-axis is vertically downward. The overhead-view satellite images are approximated as a parallel projection. The projection of 3D real-world objects onto a satellite image is formulated as:

$$\begin{pmatrix} u^s \\ v^s \end{pmatrix} = \begin{pmatrix} 1/\gamma & 0 & c^s \\ 0 & 1/\gamma & c^s \end{pmatrix} \begin{pmatrix} x^s \\ y^s \\ 1 \end{pmatrix}, \tag{1}$$

where (c^s, c^s) is the center of the satellite image and γ is the meter-per-pixel ratio.

$$\gamma = \tilde{r}_{\text{earth}} \times \frac{\cos(\tilde{L} \times \frac{\pi}{180^\circ})}{2^{\tilde{z}} \times \tilde{s}}, \tag{2}$$

where $\tilde{r}_{\text{earth}} = 156543.03392$ is the earth radius, $\tilde{L}$ is the latitude, $\tilde{z} = 18$ and $\tilde{s} = 2$ is the zoom factor and the scale of Google Maps [17], respectively. The projection of ground-view image is formulated as:

$$[u^g \ v^g \ ]^\top \propto \mathbf{K}[x^g \ y^g \ z^g]^\top, \tag{3}$$

where $\mathbf{K}$ is the intrinsic matrix of the ground camera. $[x^g \ y^g \ z^g]^\top$ is transformed from LiDAR domain to camera domain through the extrinsic matrix $\mathbf{M}_{lidar \to g}$. The alignment between the ground and satellite 3D coordinate systems is calculated using:

$$[x^s \ y^s \ z^s]^\top = \mathbf{M}_{gps \to s} \mathbf{M}_{g \to gps} [x^g \ y^g \ z^g \ 1]^\top, \tag{4}$$

where $\mathbf{M}_{g \to gps}$ is the extrinsic matrix from the camera pose to the GPS pose, given by calibration. $\mathbf{M}_{gps \to s}$ is the extrinsic matrix from the GPS pose to satellite coordinates, including three rotations: roll angle η, pitch angle ϑ and yaw angle θ; three translations: the lateral translation ϕ, the longitudinal translation φ, and height translation fixed to GPS height. During pose optimization, roll and pitch are static, while yaw is optimized. We look up sparse ground features and attention by projecting 3D points onto the correspondence views. Fig. 3 depicts the projection of 3D points onto the ground-view and satellite-view images. The projection points on satellite images depends on the pose of the query camera (ϕ, φ and θ to be estimated).

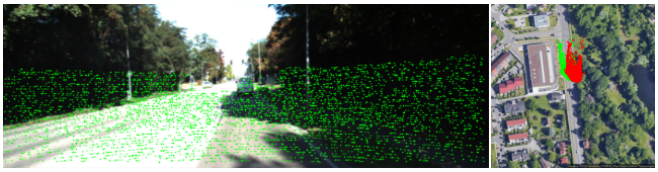


Fig. 3: (left) The visualizations of the projections of the 3D LiDAR points on the ground-view query image (green). (right) The visualizations of the projections of the 3D LiDAR points on the satellite reference image using the initial pose (red) and the ground truth pose (green), respectively.

Residual $r[p]_{\mathbf{P}}$ is calculated by subtracting sparse features across two views.

$$r_{\mathbf{P}i} = F^s[(u_i^g, v_i^g)_{\mathbf{P}}^\top] - F^g[(u_i^s, v_i^s)^\top] \in \mathbb{R}^c, \tag{5}$$

where $[\cdot]$ is a lookup with sub-pixel interpolation. c represents the feature dimension. The point weights $w_{\mathbf{P}}$ are the product of pixel-level attention across two views, for each 3D point i, we obtain:

$$w_{\mathbf{P}i} = A^s[(u_i^s, v_i^s)_{\mathbf{P}}^\top] \cdot A^g[(u_i^g, v_i^g)^\top] \in [0, 1], \tag{6}$$

where $\cdot$ is element-wise product.

C. Recursive Pose Refine Branch

Having residuals and point weights, we use LM algorithm [38], [39] to calculate the optimal solution of the vehicle pose $\mathbf{P}_{pre}$. The LM algorithm is solved by Cholesky decomposition in our approach. We follow [31] to optimize pose by successively utilizing features on each level, beginning with the coarsest level and initializing each level with the previous level's outcome. The Jacobian is defined as:

$$\mathbf{J}_i^\delta = \frac{\partial r_{\mathbf{P}i}}{\partial \delta} = \frac{\partial F^s}{\partial (u_i^s, v_i^s)_{\mathbf{P}}} \frac{\partial (u_i^s, v_i^s)_{\mathbf{P}}}{\partial (x_i^s, y_i^s)_{\mathbf{P}}} \frac{\partial (x_i^s, y_i^s)_{\mathbf{P}}}{\partial \delta}, \tag{7}$$

where δ represents an update of each element in the 3-DoF pose. The weight matrix $\mathbf{W}$ is constructed by stacking all

Fig. 4: The processing of pose optimization. RPRB optimizes the pose from an initial pose (red) towards the ground truth pose (green). The final estimation pose (blue). The color of the estimated pose during the process changes from red to blue.

points' weights to matrix diagonal:

$$\mathbf{W} = \text{Diag}(w_{\mathbf{P}i}\rho'), \tag{8}$$

where ρ' is a derivative of the robust cost function ρ [40]. The update is calculated by damping the Hessian matrices $\mathbf{H} = \mathbf{J}^\top\mathbf{W}\mathbf{J}$ and solving the linear system as (9).

$$\delta = -(\mathbf{H} + \lambda\ \text{diag}(\mathbf{H}))^{-1}\mathbf{J}^\top\mathbf{W}\Upsilon, \tag{9}$$

where $\Upsilon \in \mathbb{R}^{n\times c}$ is a matrix stacked of the residual $r_{\mathbf{P}i}$ described in (5), n is the number of 3D LiDAR points. λ is the damping factors [31].

The number of LM iterations is predetermined as 20 throughout the training process. During the test, the LM solver will stop when the update of all 3-DoF is less than 0.01. We adopt a typical pose re-projection loss on the optimized pose. The re-projection loss is formulated as:

$$L_{RPRB}(\mathbf{P}_{pre}) = \sum_i \|[u_i^s\ v_i^s]_{\mathbf{P}_{pre}} - [u_i^s\ v_i^s]_{\mathbf{P}_{gt}}\|_2^2, \tag{10}$$

where $()_{\mathbf{P}_{pre}}$ is 2D projection coordinates by the estimated pose, and $()_{\mathbf{P}_{gt}}$ is that by the ground truth pose. An example of the pose optimization process is shown in Fig. 4.

D. Pose Aware Branch

The PAB is designed to distinguish the correct pose from the erroneous ones in the feature space. This is achieved with a soft margin triplet loss [14], defined as:

$$L_{PAB} = log(1 + e^{\alpha(1-\frac{\text{Dis}(\mathbf{P}_{init}))}{\text{Dis}(\mathbf{P}_{gt})}}), \tag{11}$$

α is a hyper-parameter that is empirically set to 10. Dis$(\cdot)$ is a weighted distance that describes the distance of sparse features between ground and satellite views, formulated as:

$$\text{Dis}(\mathbf{P}) = \sum_i w_{\mathbf{P}i}\rho(\|r_{\mathbf{P}i}\|_2^2), \tag{12}$$

where i is the index of 3D points, ρ is a robust cost function [40]. $r_{\mathbf{P}i}$ is the residual of 3D point i. $w_{\mathbf{P}i}$ is the point weight. Here, we use the initial pose as an erroneous pose. The initial pose can not be treated as a wrong pose if the initial pose is close to the ground truth. So we design a hyper-parameter weight β for the triplet loss. We adopt PAB only if the re-projection error between the initial pose and ground truth pose $L_{RPRB}(\mathbf{P}_{init})$ is larger than a threshold that is empirically set to 10. We also set a top-bound which is empirically set to 50, to balance the impact of PAB.

$$L = L_{RPRB}(\mathbf{P}_{pre}) + \beta \times L_{PAB}, \tag{13}$$

where

$$\beta = \begin{cases} 0 & L_{RPRB}(\mathbf{P}_{init}) < 10 \\ L_{RPRB}(\mathbf{P}_{init}) & 10 \le L_{RPRB}(\mathbf{P}_{init}) \le 50 \\ 50 & L_{RPRB}(\mathbf{P}_{init}) > 50 \end{cases}. \tag{14}$$

IV. Dataset

We evaluate the proposed method in two standard autonomous driving datasets: KITTI [15] and Ford Multi-AV Seasonal Dataset [16]. We construct KITTI-CVL and FordAV-CVL datasets by collecting the spatial-consistent satellite counterparts from Google Map [17] according to the provided GPS tags. More specifically, we find the large region covering the vehicle trajectory and uniformly partition the region into overlapping satellite image patches. Each satellite image patch has a resolution of $1,280 \times 1,280$ pixels, amounting to about 0.2m per pixel of KITTI-CVL and 0.22m per pixel of FordAV-CVL datasets. We discovered that the satellite images obtained from Google Maps sometimes shift slightly. In FordAV-CVL datasets, the satellite view sometimes does not match the ground view using the ground truth pose. After checking the six Ford Multi-AV Seasonal trajectories, we chose the 'log4' trajectory for method evaluation, as 'log4' has the best satellite view alignment. This misalignment is less severe in the KITTI-CVL dataset. However, KITTI-CVL suffers from temporal misalignment. The camera images were taken over ten years ago, and some images are unmatched by those currently obtained satellite images due to the environment change. We used all trajectories but removed some unmatched images. In future practices, the misalignment of satellite maps can be avoided by using more accurate commercial satellite maps. We use images obtained by the front left camera from both datasets as our query inputs.

Training, Validation and Test Sets. The KITTI [15] data contains various trajectories captured at different times, with little overlap in the trajectories captured. Our validation and training data are from the same trajectory because the validation sets are used to select the best-performing model during training. In contrast, the test sets are from different trajectories for the generalization ability evaluation. For Ford Multi-AV Seasonal [16], each trajectory has been experienced three times with different weather, lighting, and traffic conditions. We aim to train our approach on one trajectory and test on the same trajectory but at a different time under different conditions.

V. Experiments

Metrics. Our goal is to estimate the 3-DoF pose. we report the median errors in lateral and longitudinal translation (m), yaw rotation (°) errors, and also the localization recall under

thresholds (0.25m, 0.5m, 1m, 2m, 1°, 2°, 4°). Since the satellite image is about 0.2m per pixel, We choose 0.25m shift error as the minor error range.

Implementation Details. Because RTK signals have already corrected the provided GPS tags in both KITTI [15] and Ford Multi-AV Seasonal [16] datasets, we use them as ground truth pose. Unless specifically stated, the initial pose is randomly sampled under 30° yaw angle errors and 10m lateral and longitudinal shifts based on the provided GPS throughout the experiments. The LiDAR data in KITTI [15] have been synchronized with camera images, we randomly sample 5000 [1] valid raw LiDAR points for each ground-view image processing. Raw LiDAR data is not synchronized with camera images in Ford Multi-AV Seasonal Dataset [16]. So, we randomly sample 5000 3D points that overlap with the camera image from the 3D map instead. VGG19 pre-trained on ImageNet [41] is adopted as an encoder to construct our U-Net. We adopt batch size $b = 3$ and Adam optimizer [42] with a learning rate of 10^{-5}.

Inference Speed. The processing time of the GaFE is around 150ms. The optimization process is executed for 20 iterations at each level, and it takes about 200ms in total.

Comparison with Image Retrieval Method DSM [43]. In order to adapt DSM [43] method to our task, we retrieve the ground-view image in a subset of satellite images. The subset of satellite images is constructed by cropping one big correspondence satellite image with 0.25m center shift on both u^s-axis and v^s-axis. For example, with 5m×5m shift range, we need to crop 20×20 $(5/0.25) = 400$ satellite images as reference images for a ground-view query image. Then, we feed the one ground-view image and 400 cropped satellite images into the DSM network to retrieve the most similar satellite image. We calculate the lateral and longitudinal distance between this retrieved satellite image center and the ground truth pose. The evaluation results are reported in Tab. I. It is clear that our method outperforms DSM [43] with a large margin. When reference images are close to each other, it is hard for image retrieval methods to distinguish image-level differences.

Comparison with Fine-grained Localization Method HighlyAccurate [37]. We train HighlyAccurate [37] model in the same setting. The evaluation results are reported in Tab. I. Our method significantly outperforms HighlyAccurate [37] in both KITTI-CVL and FordAV-CVL datasets, especially on longitudinal estimation, even though our memory usage (4736MB) is less than HighlyAccurate (6445MB). HighlyAccurate [37] does not ignore dynamic objects. It constructs geometric correspondence on the assumption that all pixels of camera images lie on the ground plane. This unreasonable assumption limits its performance. Besides, our approach provides correct geometric correspondence across 3D LiDAR points. The sparse feature alignment is reliable and efficient. To improve robustness and accuracy, we also adopt attention maps to reduce the impact of dynamic objects.

Generality. Our method demonstrated good generalization capabilities for new scenes, as shown in Tab. I. "FordAV-CVL(test)" images are from the same trajectories but different drives of the training dataset, with different environmental conditions, e.g., different weather and viewpoints. "KITTI-CVL(test)" images are from different trajectories (unseen since). "Cross-Datasets" is more challenging. We train the model in "KITTI-CVL" and test it on "FordAV-CVL(test)". In addition to different trajectories, the camera setting is also different. We compare our cross-datasets generality with HighlyAccurate [37], and the result illustrates our generalization to strongly differing scenes and settings due to the benefits from sparse feature alignment and attention map mechanisms.

Attention Map Visualization. We demonstrate attention maps in Fig. 5, which is helpful for us to discover which cues are useful or detrimental for localizing in ground-view and overhead-view images. Our network successfully extracts semantic features, e.g., road masks and edges, the boundaries of buildings and poles, etc., which contribute greatly to vehicle localization. Moreover, moving objects and repeated objects, e.g., vehicles and leaves of trees, with negative impact, are ignored. For the ground view image, it is noteworthy that the sky is assigned a high score because it is beyond the height range of the LiDAR points and is

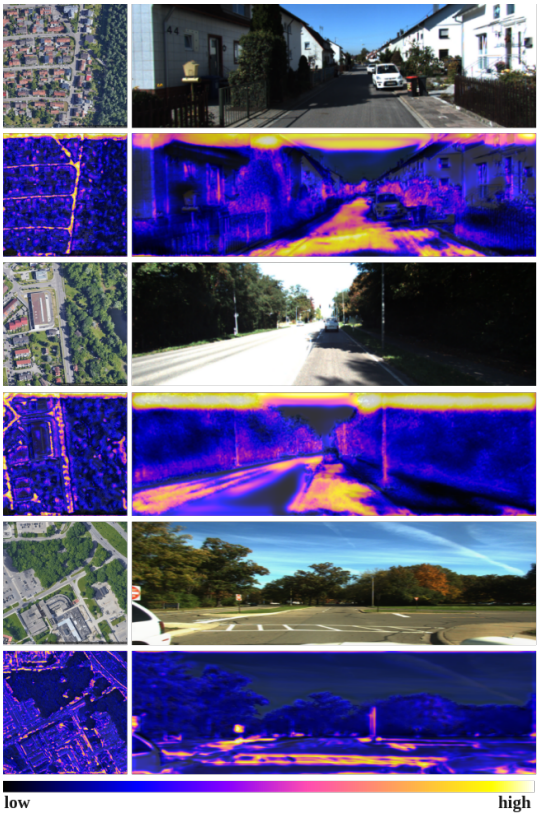


Fig. 5: Illustration of satellite, ground-view images, and their corresponding attention maps. 1^{st} and 2^{nd} are from the KITTI-CVL dataset. 3^{rd} is from the FordAV-CVL dataset.

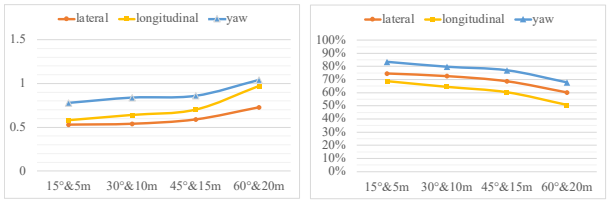
[1] We chose this sample size for training based on our experience. We have since discovered that increasing the number of points beyond this threshold does not result in significant improvements.

TABLE I: Cross-view Methods Comparison

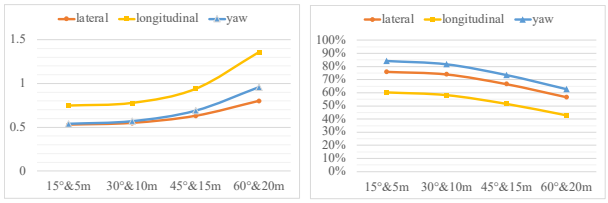
		Lateral					Longitudinal					Yaw			
		median↓	0.25m↑	0.5m↑	1m↑	2m↑	median↓	0.25m↑	0.5m↑	1m↑	2m↑	median↓	1° ↑	2° ↑	4° ↑
KITTI-CVL (evaluation)	**DSM[43]**	2.62	10.09	15.41	25.89	42.36	3.66	9.96	13.81	24.04	35.59	11.87	3.92	9.18	18.32
	HighlyAccurate[37]	0.63	21.73	41.24	68.38	87.51	1.97	7.55	14.34	27.35	50.62	1.40	38.21	62.21	81.29
	Ours	**0.22**	**54.38**	**82.54**	**95.35**	**97.83**	**0.23**	**52.03**	**77.95**	**92.02**	**96.33**	**0.46**	**81.83**	**95.01**	**98.65**
KITTI-CVL (test)	**DSM[43]**	3.48	8.51	13.97	23.44	40.61	4.97	7.67	11.02	20.03	30.48	12.73	3.13	8.29	17.44
	HighlyAccurate[37]	0.83	16.51	32.05	57.65	83.11	2.01	7.14	14.11	27.41	49.94	1.82	29.83	53.41	76.51
	Ours	**0.54**	**25.59**	**46.26**	**72.63**	**89.78**	**0.64**	**21.91**	**41.22**	**64.47**	**80.37**	**0.85**	**56.05**	**79.70**	**90.89**
FordAV-CVL (test)	**DSM[43]**	3.86	8.23	12.47	18.93	29.24	5.01	6.20	10.25	16.39	27.28	12.03	3.76	8.92	17.36
	HighlyAccurate[37]	0.84	16.56	31.31	57.64	85.45	1.82	7.11	13.87	28.53	53.64	1.83	30.74	53.08	78.40
	Ours	**0.55**	**24.83**	**45.90**	**74.06**	**89.14**	**0.78**	**18.72**	**34.11**	**58.26**	**75.44**	**0.57**	**66.76**	**81.78**	**90.50**
Cross-Datasets KITTI→FordAV	**HighlyAccurate[37]**	3.17	4.02	8.45	16.76	33.59	3.11	4.56	8.57	17.09	32.86	6.59	8.15	16.28	32.29
	Ours	**1.27**	**10.93**	**21.04**	**40.84**	**65.58**	**1.63**	**8.59**	**16.97**	**32.28**	**57.75**	**1.67**	**32.65**	**56.20**	**73.45**

not sampled in the algorithm. As the sky's confidence score is not actively monitored during training, it may retain an initial high value. However, given that the sky does not contribute to pose optimization, its high confidence score will not have any impact on vehicle localization. Additionally, our method employs a confidence mechanism that eliminates the requirement for simultaneous acquisition of satellite and ground-view imagery. This mechanism automatically reduces the impact of dynamic objects and prioritizes more stable objects that are more useful in vehicle pose estimation.

Performance under Different Initial Poses. We tested our method using the initial pose with a more extensive and challenging bound. The results are shown in Fig. 6. It shows that our method not only performs well with the initial poses under the same level when training the network but is also robust when using the initial pose under a larger shift. To be more specific, our approach is robust and achieves a satisfactory accuracy with initial raw pose under 60° yaw angle errors and 20m lateral and longitudinal shifts.



(a) KITTI-CVL(test)



(b) FordAV-CVL(test)

Fig. 6: (left) Median lateral and longitudinal translation (m) and yaw rotation (°) errors under different initial poses. (right) Recall under lateral and longitudinal translation error 1m and yaw rotation errors 2°.

VI. Ablation Study

Shared Weights between the Two Views. One of the key challenges in cross-view visual localization is to extract features that are robust to the appearance gap between ground-view and overhead-view images. Satellite and ground-view images have different resolutions, different viewpoints, and various camera intrinsic. That is why most cross-view methods [11], [9], [10], [43], [34], [13], [37] use siamese architecture without shared weights to do metric learning. In contrast, [31] shares weights between two branches and adopts attention maps to mitigate domain shift. We evaluate both methods, and the result of the KITTI-CVL (test) is shown in Tab. II. The latter (full) performs much better than the former (w/o SW). A potential reason is that the shared weights branches help the learned features be in the same domain. Thus, shared weights facilitate more accurate pose estimation.

TABLE II: Ablation Study

	Lateral		Longitudinal		Yaw	
	median↓	1m↑	median↓	1m↑	median↓	2° ↑
w/o SW	0.84	56.25	1.19	44.32	1.51	58.88
w/o PAB	0.59	69.64	0.68	62.89	0.86	78.74
w/o β	0.58	70.10	0.68	64.04	0.85	79.20
full	**0.54**	**72.63**	**0.64**	**64.47**	**0.85**	**79.70**

Effectiveness of PAB. We designed two branches of objective functions. The RPRB encourages the predicted pose close to the ground truth. The PAB discriminates residuals between the correct and incorrect pose. Unlike RPRB which passes gradient from pose to feature extractor weights, the gradient backpropagates of PAB are directly from triplet loss to feature extractor weights. The PAB allows the network to learn the pose-aware features before RPRB can estimate a proper pose. We adopt a hyper-parameter PAB loss weight β to balance the effect of PAB. The results of the ablation studies on β and PAB of KITTI-CVL(test) are shown in Tab.II. They all contribute to the performance of SIBCL.

VII. Conclusions

Our paper presents a robust and novel geometry-driven correspondence learning approach for 3-DoF camera pose estimation. SIBCL is the first cross-view approach capable of accomplishing localization with a median position error below 1 meter and a median orientation error below 1°, without relying on a high-definition map. It makes maximum use of satellite images for fine-grained localization. Furthermore, it generalizes well to new scenes and thus can be used as an interpretable prior and adapted to new scenes after brief fine-tuning. In the future, we will consider using multi-cameras and low-cost sensors. We believe our work may lead to reliable, accurate, and low-cost vehicle localization systems.

REFERENCES

[1] L. Xiong, R. Kang, J. Zhao, P. Zhang, M. Xu, R. Ju, C. Ye, and T. Feng, "G-vido: A vehicle dynamics and intermittent gnss-aided visual-inertial state estimator for autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[2] C. McManus, W. Churchill, W. Maddern, A. D. Stewart, and P. Newman, "Shady dealings: Robust, long-term visual localisation using illumination invariance," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 901–906.

[3] G. Pascoe, W. Maddern, and P. Newman, "Direct visual localisation and calibration for road vehicles in changing city environments," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 9–16.

[4] G. Pascoe, W. P. Maddern, and P. Newman, "Robust direct visual localisation using normalised information distance." in *BMVC*, 2015, pp. 70–1.

[5] W. Maddern, G. Pascoe, and P. Newman, "Leveraging experience for large-scale lidar localisation in changing cities," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1684–1691.

[6] C. Le Gentil, T. Vidal-Calleja, and S. Huang, "In2lama: Inertial lidar localisation and mapping," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6388–6394.

[7] D. Droeschel and S. Behnke, "Efficient continuous-time slam for 3d lidar-based online mapping," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5000–5007.

[8] A. Vora, S. Agarwal, G. Pandey, and J. McBride, "Aerial imagery based lidar localization for autonomous vehicles," 2020. [Online]. Available: https://arxiv.org/abs/2003.11192

[9] Y. Shi, X. Yu, L. Liu, T. Zhang, and H. Li, "Optimal feature transport for cross-view image geo-localization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 11 990–11 997.

[10] Y. Shi, L. Liu, X. Yu, and H. Li, "Spatial-aware feature aggregation for image based cross-view geo-localization," in *Advances in Neural Information Processing Systems*, 2019, pp. 10 090–10 100.

[11] S. Hu, M. Feng, R. M. Nguyen, and G. H. 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*, 2018, pp. 7258–7267.

[12] L. Liu and H. Li, "Lending orientation to neural networks for cross-view geo-localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5624–5633.

[13] A. Toker, Q. Zhou, M. Maximov, and L. 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*, 2021, pp. 6488–6497.

[14] Q. Qian, L. Shang, B. Sun, J. Hu, H. Li, and R. Jin, "Softtriple loss: Deep metric learning without triplet sampling," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6450–6458.

[15] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[16] S. Agarwal, A. Vora, G. Pandey, W. Williams, H. Kourous, and J. McBride, "Ford multi-AV seasonal dataset," *The International Journal of Robotics Research*, vol. 39, no. 12, pp. 1367–1376, sep 2020. [Online]. Available: https://doi.org/10.1177%2F0278364920961451

[17] (2022). [Online]. Available: https://developers.google.com/maps/documentation/maps-static/overview

[18] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[19] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[20] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European conference on computer vision*. Springer, 2014, pp. 834–849.

[21] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[22] R. C. DuToit, J. A. Hesch, E. D. Nerurkar, and S. I. Roumeliotis, "Consistent map-based 3d localization on mobile devices," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE Press, 2017, p. 6253–6260. [Online]. Available: https://doi.org/10.1109/ICRA.2017.7989741

[23] E. Jones and S. Soatto, "Visual-inertial navigation, localization and mapping: A scalable real-time large-scale approach," *Intl. J. of Robotics Res*, vol. 6, 2011.

[24] S. Lynen, T. Sattler, M. Bosse, J. A. Hesch, M. Pollefeys, and R. Siegwart, "Get out of my lab: Large-scale, real-time visual-inertial localization." in *Robotics: Science and Systems*, vol. 1, 2015, p. 1.

[25] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular slam with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.

[26] S. Hausler, M. Xu, S. Garg, P. Chakravarty, S. Shrivastava, A. Vora, and M. Milford, "Improving worst case visual localization coverage via place-specific sub-selection in multi-camera systems," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 112–10 119, 2022.

[27] T. Schneider, M. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart, "maplab: An open framework for research in visual-inertial mapping and localization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1418–1425, 2018.

[28] A. G. Vora, S. Agarwal, J. N. Hoellerbauer, and F. Shaik, "High definition 3d mapping," June 6 2019, uS Patent App. 15/831,295.

[29] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt, "Scalable 6-dof localization on mobile devices," in *European conference on computer vision*. Springer, 2014, pp. 268–283.

[30] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[31] P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, and T. Sattler, "Back to the Feature: Learning robust camera localization from pixels to pose," in *CVPR*, 2021. [Online]. Available: https://arxiv.org/abs/2103.09213

[32] L. Von Stumberg, P. Wenzel, N. Yang, and D. Cremers, "Lm-reloc: Levenberg-marquardt based direct visual relocalization," in *2020 International Conference on 3D Vision (3DV)*. IEEE, 2020, pp. 968–977.

[33] R. W. Wolcott and R. M. Eustice, "Visual localization within lidar maps for automated urban driving," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 176–183.

[34] S. Zhu, T. Yang, and C. 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*, 2021, pp. 3640–3649.

[35] I. D. Miller, A. Cowley, R. Konkimalla, S. S. Shivakumar, T. Nguyen, T. Smith, C. J. Taylor, and V. Kumar, "Any way you look at it: Semantic crossview localization and mapping with lidar," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2397–2404, 2021.

[36] F. Fervers, S. Bullinger, C. Bodensteiner, M. Arens, and R. Stiefelhagen, "Continuous self-localization on aerial images using visual and lidar sensors," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 7028–7035.

[37] Y. Shi and H. Li, "Beyond cross-view image retrieval: Highly accurate vehicle localization using satellite image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

[38] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of applied mathematics*, vol. 2, no. 2, pp. 164–168, 1944.

[39] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[40] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust statistics: the approach based on influence functions*. John Wiley & Sons, 2011, vol. 196.

[41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[43] Y. Shi, X. Yu, D. Campbell, and H. Li, "Where am i looking at? joint location and orientation estimation by cross-view matching," in

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 4064–4072.