View Consistent Purification for Accurate Cross-View Localization

Shan Wang[1,2] Yanhao Zhang[1] Akhil Perincherry[3] Ankit Vora[3] Hongdong Li[1]
[1]Australian National University [2]Data61, CSIRO [3]Ford Motor Company

Abstract

This paper proposes a fine-grained self-localization method for outdoor robotics that utilizes a flexible number of onboard cameras and readily accessible satellite images. The proposed method addresses limitations in existing cross-view localization methods that struggle to handle noise sources such as moving objects and seasonal variations. It is the first sparse visual-only method that enhances perception in dynamic environments by detecting view-consistent key points and their corresponding deep features from ground and satellite views, while removing off-the-ground objects and establishing homography transformation between the two views. Moreover, the proposed method incorporates a spatial embedding approach that leverages camera intrinsic and extrinsic information to reduce the ambiguity of purely visual matching, leading to improved feature matching and overall pose estimation accuracy. The method exhibits strong generalization and is robust to environmental changes, requiring only geo-poses as ground truth. Extensive experiments on the KITTI and Ford Multi-AV Seasonal datasets demonstrate that our proposed method outperforms existing state-of-the-art methods, achieving median spatial accuracy errors below 0.5 *meters along the lateral and longitudinal directions, and a median orientation accuracy error below* 2° [1].

1. Introduction

Accurate self-localization is a fundamental problem in mobile robotics, particularly in the context of autonomous driving. While Global Positioning System (GPS) is a widely adopted solution, its accuracy hardly meets the stringent requirements of autonomous driving [20]. Real-Time Kinematic (RTK) positioning systems provide an alternative by correcting GPS errors, but their implementation is hindered by the need for signal reference stations [13], rendering them an expensive solution. On the other hand, odometry [18, 4, 37, 32] or simultaneous localization and mapping (SLAM) [17, 11, 25, 32] methods can generate accurate short-term trajectories, however, they experience drift accumulation over time that can only be alleviated through loop closures if the agent's trajectories overlap. Lastly, other self-localization techniques [35, 15, 31, 21] that rely on a pre-constructed 3D High Definition (HD) maps face limitations in terms of the extensive time and resources required for map acquisition and maintenance.

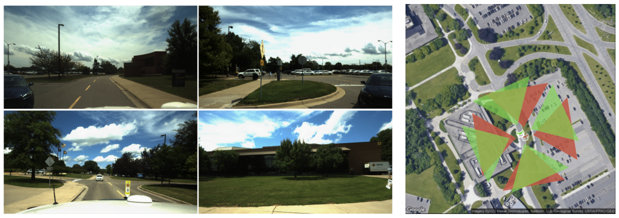

(a) *Query* (b) *Reference*

Figure 1: (a) Query ground view (onboard camera) images (front, rear, left, and right). (b) reference satellite image. The initial and ground truth poses, and FoV of cameras are shown in (red) and (green), respectively.

Using off-the-shelf satellite images as ready-to-use maps to achieve cross-view localization brings an alternative and promising way for low-cost localization. However, due to the significant disparity between overhead views captured by satellites and views seen by robots, cross-view localization is more challenging than traditional methods. To address this, it is crucial to purify view-consistent features that can support the localization process. Furthermore, satellite views can be captured at different times, leading to variations in seasonal and temporal conditions. The cross-view consistent purification can also minimize the impact of moving and seasonal objects.

Most previous cross-view localization methods [24, 10, 14, 29, 23, 38] approach the task as an image retrieval problem, leading to coarse localization accuracy that is inferior to commercial GPS which can achieve an error of up to 4.9 meters in open sky conditions [30]. In contrast, our method utilizes a coarse pose that is easily obtainable from the Autonomous Vehicles system, to estimate the fine-grained 3-DoF (lateral, longitudinal, yaw) pose of the robot. This is accomplished through visual cross-view matching, utilizing ground-view images captured by onboard cameras and a

[1] Our project page is https://shanwang-shan.github.io/PureACL-website/

spatially-consistent satellite map. Additionally, our method supports multiple camera inputs, which extend the field of view of the query robot. The setting is illustrated in Fig. 1.

Our fine-grained visual localization method utilizes sparse (keypoint) feature matching, a departure from prior methods that rely on dense feature matching. To reduce the inherent ambiguity in purely visual matching, the method incorporates a camera intrinsic and extrinsic aware spatial embedding. Homography transformation is used to establish correspondences between the two views. An on-ground confidence map is employed to ensure the validity of the transformation and eliminate off-the-ground objects. Additionally, a view consistency confidence map is utilized to mitigate the impact of moving objects and viewpoint variation. The localization process begins with the extraction of spatially aware deep features and the generation of view-consistent, on-ground confidence maps for both views. View-consistent key points are then detected from the ground view confidence map and matched with their corresponding points in the satellite view. The optimal pose is determined through an iterative search using a differentiable Levenberg-Marquardt (LM) algorithm.

Using Google Maps [8] as the satellite view, we evaluate our method on two datasets: the Ford Multi-AV Seasonal (FMAVS) [1] and the KITTI Datasets [7]. The results demonstrate the superiority of our proposed method, achieving mean localization error of less than $\{0.14m, 3.57°\}$ on KITTI with one front-facing onboard camera, and less than $\{0.88m, 0.74°\}$ on FMAVS with four surrounding onboard cameras.

We summarize our contributions as below:

- the first sparse visual-only cross-view localization method that estimates accurate pose with low spatial and angular errors.
- a view-consistent on-ground key point detector that reduces the impact of dynamic objects and viewpoint variations, as well as removes off-the-ground objects.
- a spatial embedding that fully utilizes camera intrinsic and extrinsic information to improve the extraction of spatially aware visual features.
- a multi-camera fusion approach that significantly improves localization accuracy.

2. Related work

Depth Aware Accurate Cross-view Localization. The task of accurate cross-view localization has gained attention in recent years. Researchers have mainly focused on developing solutions for Radar and LiDAR cross-view localization as depth information helps in aligning the ground and satellite perspectives. RSL-Net [26] estimates the robot pose by registering Radar scans on a satellite image. This method was later extended to a self-supervised learning framework in [28]. Another work [27] matches the top-down representation of a LiDAR scan with 2D points detected from satellite images. These methods have limitations and are only effective in environments with strong prior structure knowledge, failing in general, non-urban environments. [2] performs localization on bird's eye view (BEV) LiDAR intensity maps using deep feature matching between LiDAR scan and the intensity map. [34] extends this method by incorporating compressed binary maps. Hybrid sensor solutions have also been explored, such as in [16] where an aerial robot achieves global localization through the use of egocentric 3D semantically labelled LiDAR, IMU, and visual information. CSLA [6] and SIBCL [33] extract visual features from ground and satellite images and use LiDAR points to establish correspondence between the two views. CSLA [6] aims to estimate 2-DoF translation, while SIBCL [33] aims to estimate 3-DoF pose, including an additional orientation. All these methods critically rely on depth information to build the correspondence across the two views. In contrast, our method is a visual-only solution that aims to achieve comparable localization accuracy using cheaper commodity sensors.

Visual Accurate Cross-view Localization. Most visual-only cross-view localization methods rely on homography transformations of the ground plane, as they lack reliable depth information. [36] aims to estimate 2-DoF translation using similarity matching and produces a dense spatial distribution to address localization ambiguities. HighlyAccurate [22] projects satellite features into the ground view and optimizes the robot pose through dense feature matching. One of its drawbacks is the limited ability to effectively eliminate outliers, such as noise caused by off-the-ground objects (which violates the assumption of homography transformation of the ground plane) and dynamic objects. As a result, their overall performance is limited. In contrast, our method constructs geometric correspondences across sparse view-consistent on-ground keypoints, ensuring that the pose estimation is based on accurate correspondences leading to improved precision.

3. Our Method

Our work aims to achieve fine-grained cross-view localization by accurately estimating the 3-DoF pose, denoted by $\mathbf{P}_{pred} = \{\phi_{pred}, \varphi_{pred}, \theta_{pred}\}$, where ϕ and φ represent lateral and longitudinal translations, respectively, and θ is the yaw angle. We are given a coarse initial pose $\mathbf{P}_{init} = \{\phi_{init}, \varphi_{init}, \theta_{init}\}$, a reference satellite view image I^s, and a set of ground-view images $I^g = \{I^i\}_{i=1}^N$ captured by onboard cameras, where N is the total num-

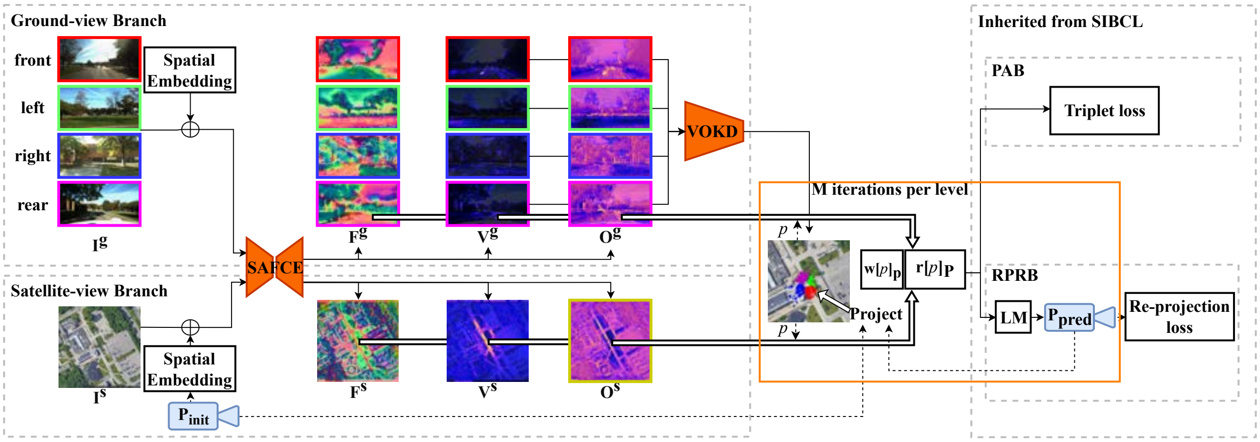


Figure 2: Overview of PureACL. SAFCE is used to produce feature maps (F), view-consistent confidence maps (V), and on-ground confidence maps (O) separately for satellite and ground-view images. The VOKD fuses the confidence maps and identifies the top-k confident features from the ground-view images and their corresponding features on the satellite feature maps. Sub-pixel interpolation is used to lookup point features ($F[p]$ from F) and their weights ($w[p]$ from $V \otimes O$). The residual between the two views ($r[p]_{\mathbf{P}} = F^s[p]_{\mathbf{P}} - F^g[p]$) and the point weights ($w[p]_{\mathbf{P}} = w^s[p]_{\mathbf{P}} \times w^g[p]$) are fed to the RPRB for subsequent pose optimization. The olive outline indicates that the O^s disables gradient backpropagation while red, green, blue and magenta outlines and points represent the front, left, right and rear views, respectively.

ber of onboard cameras [2]. An overview of the proposed PureACL is shown in Fig. 2. It builds upon three innovative modules: 1) Spatially Aware Feature and Confidence Extractor (SAFCE) (Sec. 3.2), 2) View-consistent On-ground Keypoint Detector (VOKD) (Sec. 3.3), and 3) Multi-camera Fusion (Sec. 3.4). Additionally, our approach utilizes two branches of objective functions inherited from the SIBCL method [33]: the Pose-Aware Branch (PAB) and the Recursive Pose Refine Branch (RPRB). In the following sections, we provide a detailed explanation of each module.

3.1. Preliminary

For completeness, we provide a brief description of the inherited PAB and RPRB. The PAB utilizes a triplet loss [19] that encourages accurate pose (ground truth) and penalizes incorrect (initial) poses by differentiating the residual between the ground truth and initial pose. Specifically, we compute the loss as follows:

$$L_{triplet} = log(1 + e^{\alpha(1 - \frac{\sum_p w[p]_{\mathbf{P}_{init}} \rho(\|r[p]_{\mathbf{P}_{init}}\|_2^2)}{\sum_p w[p]_{\mathbf{P}_{gt}} \rho(\|r[p]_{\mathbf{P}_{gt}}\|_2^2)})}), \quad (1)$$

where α is a hyper-parameter set to 10 based on experimental results, $\sum_p$ represents the sum of all key points, and ρ is a robust cost function as defined in [9].

The RPRB, on the other hand, aims to refine the initial pose iteratively using the LM algorithm to approach the ground truth pose. It starts with the coarsest level and uses features from each level successively, with each subsequent level initialized with the output of the previous level. Specifically, we update the pose as follows:

$$\delta_{t+1} = \delta_t - (\mathbf{H} + \lambda \, \mathrm{diag}(\mathbf{H}))^{-1} \mathbf{J}^\top \mathbf{W} \Upsilon, \quad (2)$$

where δ represents an individual element in the 3-DoF pose. $t \in \{1, \cdots, M \times L\}$ represents the current iteration, and M and L represent the iteration count per level and the total number of levels, respectively. The matrices Υ and $\mathbf{W}$ are formed by stacking the residuals $r[p]_{\mathbf{P}}$ and weights $w[p]_{\mathbf{P}}$, while λ is the damping factors [21]. The Jacobian and Hessian matrices are defined as follows:

$$\mathbf{J} = \frac{\partial r[p]_{\mathbf{P}}}{\partial \delta} = \frac{\partial F^s[p]}{\partial [p^s_{2D}]_{\mathbf{P}}} \frac{\partial [p^s_{2D}]_{\mathbf{P}}}{\partial \delta} \text{ and } \mathbf{H} = \mathbf{J}^\top \mathbf{W} \mathbf{J}, \quad (3)$$

where $[p^s_{2D}]_{\mathbf{P}}$ is the 2D projection of keypoints p onto the satellite image using the pose $\mathbf{P}$, as shown in the right section of Fig. 6. Finally, we supervise the optimized pose by computing the re-projection error as follows:

$$L_{reproject}(\mathbf{P}_{pred}) = \sum \|[p^s_{2D}]_{\mathbf{P}_{pred}} - [p^s_{2D}]_{\mathbf{P}_{gt}}\|_2^2. \quad (4)$$

3.2. Spatially Aware Feature/Confidence Extractor

Our approach improves the spatial embedding concept proposed in [14] by leveraging the camera's intrinsic and extrinsic parameters to obtain highly accurate spatial information. The spatial embedding $E^{g/s} \in \mathbb{R}^{h \times w \times 3}$ has 3 channels: heading, distance, and height information. The explanation of these channels is shown in Fig. 3. To incorporate additional spatial embedding information between

[2] Our method supports varying onboard camera quantities. In the experiments, we employed $N = 4$ for FMAVS and $N = 1$ for Kitti-CVL.

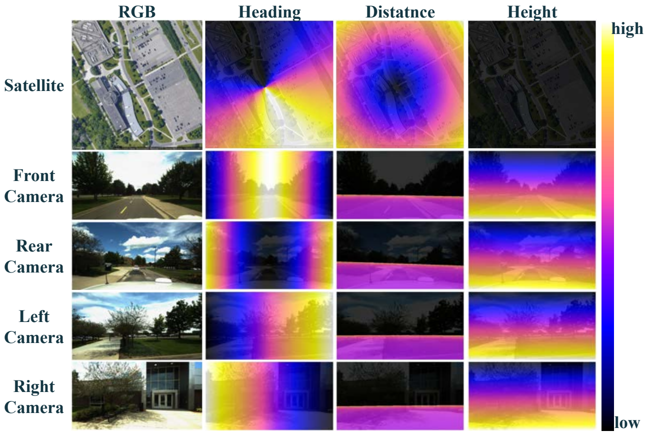


Figure 3: The Spatial Embedding. Heading is embedded using the cosine of its angle. Distance is embedded as the normalized on-ground distance from the robot, with the assumption that the pixel is lying on the ground. Height is normalized on the down axis of the ground view coordinates. For satellite images, the height is specifically set to a minimal value to indicate a top-down view.

the ground and satellite images, we transform the pixels in the onboard camera and satellite images into a common set of query world coordinates (e.g., the GPS coordinates of the robot). In this coordinate system, the x-axis corresponds to the direction of motion, the y-axis points to the right, and the z-axis points downward. To perform this transformation, we use an inverse projection formula, which is shown in Eq. 5:

$$p_{3D}^{j2g} = \mathbf{R}_{j2g}\mathbf{K}_j^{-1}(p_{2D}^{j} \oplus 1), \tag{5}$$

where $\mathbf{K}_j$ is the intrinsic matrix of camera j, which can be either an onboard camera or a satellite camera $j \in \{i_1^N, s\}$, and $\oplus 1$ concatenates 1 to generate the homogeneous coordinate. The rotation from camera j to the ground coordinate, $\mathbf{R}_{j2g}$, is obtained from the extrinsic information provided in the datasets for onboard cameras and from the initial coarse pose for the satellite camera. For onboard camera images, the 3D coordinate p_{3D}^{i2g} is a homogeneous coordinate with an unknown scale, while for satellite images, p_{3D}^{s2g} represents a world coordinate with an unknown down axis. This is because satellite images are approximated as parallel projections, and the equation for the calculating p_{2D}^{s} is given by:

$$p_{2D}^{s} = \begin{pmatrix} 1/\gamma & 0 & c_u \\ 0 & 1/\gamma & c_v \end{pmatrix} p_{3D}^{s}, \tag{6}$$

where (c_u, c_v) represents the center of the satellite image, and γ represents the meter-per-pixel ratio calculated using:

$$\gamma = \tilde{r}_{\text{earth}} \times \frac{\cos(\tilde{L} \times \frac{\pi}{180^\circ})}{2^{\tilde{z}} \times \tilde{s}}, \tag{7}$$

where $\tilde{r}_{\text{earth}} = 156543.03392$ is radius of the Earth, $\tilde{L}$ is the latitude, $\tilde{z} = 18$ and $\tilde{s} = 2$ is the zoom factor and the scale of Google Maps [8], respectively.

The heading information is embedded using the cosine value, which is symmetric to both positive and negative orientation noise. This enables distinction between 360-degree views, calculated using the x-axis ($p_{3D}^{j2g}[0]$) and y-axis ($p_{3D}^{j2g}[1]$) through trigonometric functions, as shown below:

$$E^{j}[0] = p_{3D}^{j2g}[0] / \sqrt{p_{3D}^{j2g}[0]^2 + p_{3D}^{j2g}[1]^2}. \tag{8}$$

The normalized distance embedding of ground images is obtained by assuming all pixels lie on the ground plane:

$$E^{j}[1] = \sqrt{p_{\tilde{3D}}^{j2g}[0]^2 + p_{\tilde{3D}}^{j2g}[1]^2} / \mathcal{D}, \tag{9}$$

where $\mathcal{D}$ is the maximum visible distance, set to 200 meters according to the satellite maps size and

$$p_{\tilde{3D}}^{i2g} = \frac{h_i}{p_{3D}^{i2g}[2]} \times p_{3D}^{i2g} + \mathbf{t}_{i2g} \text{ and } p_{\tilde{3D}}^{s2g} = p_{3D}^{s2g} + \mathbf{t}_{s2g}, \tag{10}$$

where h_i is the onboard camera height relative to the ground plane. For ground view images, the height embedding $E[2]$ is equal to the value along the down axis, represented as $p_{\tilde{3D}}^{i2g}[2]$. In the case of satellite images, we set the height embedding to the minimal value to indicate a top-down perspective. Fig. 4 demonstrates that our approach effectively directs greater attention towards the features located in front of the robot by leveraging spatial embedding when using only the front onboard camera.

The SAFCE employs a U-Net structure ($\mathcal{F}_\nu$) to extract the satellite and ground-view feature maps, represented as $F^j = \mathcal{F}_\nu(I^j \oplus E^j)$, where $j \in \{i_1^N, s\}$, and $\oplus$ denotes channel concatenation. The maps are then processed by a convolutional layer followed by a reverse sigmoid active function ($\mathcal{C}_\psi$) to produce view-consistent confidence maps (V^j) and on-ground confidence maps (O^j) represented as $V^j, O^j = \mathcal{C}_\psi(F^j)$. Each map has multiple resolutions, for example, $F = \{F_l \in \mathbb{R}^{h_l \times w_l \times c_l}\}_{l=1}^{L}$ ($\mathbb{R}^{h_l \times w_l}$ for V and O), where $L = 3$ is adopted in our setting. The maps are ordered from coarsest to finest level as $l = \{1, 2, 3\}$. The feature and confidence extraction from each image is performed in parallel using a shared-weight model, allowing for a flexible number of onboard cameras (N).

The view-consistent confidence map V represents the confidence of objects appearing in both satellite and ground-view images. V is used as a multiplying factor for the point weights supervised by PAB and RPRB, and is penalized through the network training for the points with high residual (indicating distinct features between the cross-view). Considering the temporal gap between the two views, V effectively filters out objects that are temporally

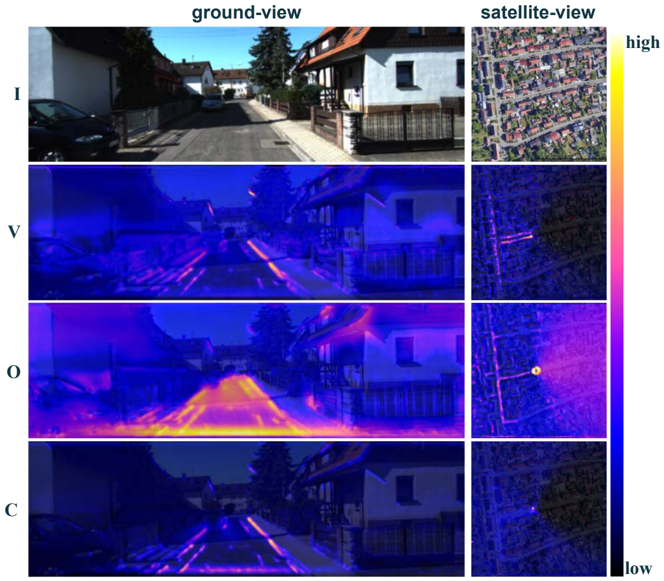


Figure 4: Illustration of confidence maps. The view-consistent confidence map (2^{nd} row) V assigns high confidence to objects that appear consistently in both ground-view and satellite images, such as road marks, curbs, and building roofs. Conversely, the confidence map assigns low confidence to temporally inconsistent objects, such as vehicles. The on-ground confidence map (3^{rd} row) O highlights only on-ground cues, such as road marks and curbs. It is noteworthy that the area behind the robot is assigned a high score due to a lack of supervision, but it does not affect localization accuracy. This is because the influence of the area is suppressed by the view-consistent confidence ($w[p]$ from $V \times O$). The fused confidence map (4^{th} row) C highlights objects that are both view-consistent and on-ground.

or seasonally inconsistent, e.g. vehicles, pedestrians, and leaves. Additionally, it highlights view consistent reference objects, including road marks, lanes, building edges, and tree roots. An example is shown in Fig. 4 (row 2). More visualizations are shown in the supplementary.

The on-ground confidence map O is designed to validate the homography transformation between the ground and satellite views. As a multiplying factor for the point weights, off-ground points that cause incorrect Geo-correspondence between the ground and satellite views, resulting in high residuals, have their on-ground confidence penalized to reduce the overall loss. Given that an incorrect height assumption in points can lead to erroneous projections on the satellite map, penalizing the satellite on-ground confidence map is not meaningful. So we only apply the backpropagation to the ground-view on-ground confidence map. An example of the learned confidence maps is shown in Fig. 4 (row 3).

3.3. View-consistent On-ground Keypoint Detector

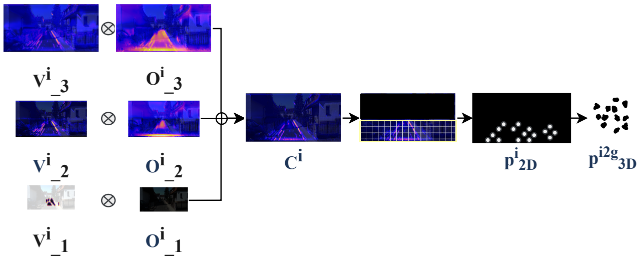


Figure 5: The pipeline of VOKD. It begins with confidence map fusion, in which all level confidence maps from the view-consistent and on-ground maps are combined to create a single map. Next, in the 2D keypoint detection step, the top part of the image is ignored to concentrate on the ground plane. Moreover, a max pooling technique is employed to avoid overly crowded keypoint detection. Finally, based on the assumption that all detected points are on the ground, their 3D ground query coordinates are calculated.

Fig. 5 illustrates the details of the proposed VOKD. The view-consistent and on-ground confidence maps of different resolutions are fused to generate the final confidence:

$$C^i = \sum_{l=1}^{L} \Xi(\mathcal{N}(V_l^i \otimes O_l^i), (h_L, w_L)), \tag{11}$$

where h_L and w_L represents the resolution of the fine level confidence map, Ξ is an interpolation function, and $\mathcal{N}$ is a min-max normalisation, and $\otimes$ represents element-wise multiplication. The bottom row of Fig. 4 demonstrates the efficacy of the fused confidence map in filtering out off-the-ground objects and emphasizing temporal stability and view consistency in cues such as road markings and curbs for the subsequent pose estimation. More visual examples can be found in the supplementary.

In order to achieve on-ground keypoint detection, our focus is limited to the area below the focal point, which corresponds to the on-ground area and is our primary interest. From this area, we select the top-K points with the highest confidence score from the fused confidence map. To avoid overcrowding of keypoints, we partition the fused confidence map into smaller patches of size 8×8 and enforce a limit of one detected keypoint per patch. This approach ensures that the selected keypoints are well-distributed across the on-ground area, thereby improving the accuracy of subsequent pose estimation. The left part of Fig. 6 displays the detected view-consistent on-ground 2D keypoints. These 2D keypoint coordinates p_{2D}^i are used to calculate their corresponding 3D ground world coordinates $p_{\tilde{3D}}^{i2g}$ through the equations Eq. (5) and Eq. (10). The right part of Fig. 6 shows the projection of these 3D coordinates onto the satellite image ($p_{2D}^s = \mathbf{K}_s(\mathbf{R}_{g2s} p_{\tilde{3D}}^{i2g} + \mathbf{t}_{g2s})$).

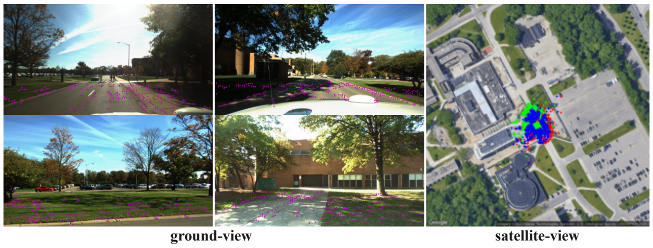


Figure 6: (left) On-ground keypoints on ground-view images and detected keypoints (magenta). (right) Projection of on-ground keypoints on the satellite image. Projection by initial pose is shown in (red), projection by predicted pose is shown in (blue), and projection on ground truth pose is shown in (green).

3.4. Multi-camera Fusion

Our method is flexible and can handle multiple cameras as input, without any restrictions on the field of view. In case there is a potential overlap between the views captured by adjacent cameras, keypoints detected in one camera may be visible in another camera as well. In such cases, we select the point feature with the highest weight:

$$w^g[p] = \max_i^N (V^i \otimes O^i)[p^i_{2D}], \tag{12}$$

$$F^g[p] = F^i[p^i_{2D}], \;\; i = \arg\max_i^N (V^i \otimes O^i)[p^i_{2D}]. \tag{13}$$

4. Datasets

To evaluate the effectiveness of the proposed method, we followed the existing methods [22, 33] and conducted experiments on two widely used autonomous driving datasets: the FMAVS dataset [1] and KITTI dataset [7]. We adopted the augmentation method proposed by [33], which involved incorporating spatially-consistent satellite images obtained from Google Maps [8] using the GPS tags provided in the datasets. The satellite images had a resolution of $1,280 \times 1,280$ pixels and a scale of 0.22m per pixel for FordAV-CVL, and 0.2m per pixel for KITTI-CVL.

In the FMAVS dataset, we utilized query images from four cameras (front left, rear right, side left, and side right) to capture the surrounding environment, providing an almost 360-degree field of view with minimal overlap. Since the KITTI dataset provides only front-facing stereo camera images, we used the images from the left camera of the stereo pair as query images. The FMAVS includes multiple vehicle traversals over a consistent route. To evaluate our proposed method, we split the three traversals of the 'Log4' trajectory into training, validation, and test sets, following the split strategy described in [33]. The KITTI dataset [7] comprises various trajectories taken at different times. To assess our model's generalization ability, we selected test sets from different trajectories based on [22].

5. Experiments

Metrics. Our objective is to estimate the 3-DoF pose, which includes lateral, longitudinal, and yaw information. We measure the accuracy of our proposed method by reporting the median and mean errors in lateral and longitudinal translations (in meters) and yaw rotation (in degrees). In addition to these metrics, we also follow the evaluation criteria outlined in [33] and report the average localization recall [3] at distances of 0.25m, 0.5m, 1m, and 2m, as well as at yaw rotation angles of $1°$, $2°$, and $4°$.

Implementation Details. In our experiments, we use an input size of 432×816 for the ground-view images in the Multi-AV Seasonal Dataset, and 384×1248 for the KITTI Dataset. RTK GPS [4] is used as the ground truth pose. We add some noise to the RTK GPS poses to generate the initial pose. Unless otherwise stated, the initial pose is randomly sampled with a yaw angle error of $\pm 15°$ and lateral, longitudinal shifts of ± 5 meters, as the accuracy of GPS is within 4.9 meters in open sky conditions [30]. We detect 256 ground keypoints from each input ground-view image. We set the batch size to $b = 3$ for training on an NVIDIA RTX 3090 GPU, and use the Adam optimizer [12] with a learning rate of 10^{-4}. The feature extractor weights are initialized with the pre-trained weights from [33], which are trained on the KITTI-CVL dataset. The weights of the confidence generator are initially randomly initialized to values near 0. Through the application of the inverse sigmoid activation function, these weights are tuned to initialize the confidence values in proximity to 50%.

Inference Speed. The SAFCE processes four query ground-view images and one satellite image in approximately 200ms. The detection time for all ground keypoints is about 3.5ms. The optimization process, which runs for 20 iterations at each of the three levels, takes a total of approximately 200ms.

Qualitative Results. We compare our method with recent state-of-the-art (SOTA) visual-only methods, CVML [36] and HighlyAccurate [22], as well as the LiDAR-visual hybrid method SIBCL [33]. We present the evaluation results on the KITTI-CVL and FordAV-CVL datasets in Tab. 1 and Tab. 2. To ensure a fair comparison, we trained HighlyAccurate [22] and SIBCL [33] under the same image resolution and initial pose noise range. Since CVML [36] is unable to accurately estimate fine-grained orientations, we only evaluated its performance in terms of location estimation. We trained their model with ground truth orientation.

Tab. 1 presents an evaluation of our method's ability to generalize to previously unseen routes in the KITTI-CVL dataset using a front camera. For translation accuracy, our method exhibits superior performance compared to SOTA

[3] The percentage of the prediction pose that is within a certain range.

[4] RTK GPS achieves an accuracy of 2 cm or better [5].

Table 1: Comparison on the KITTI-CVL dataset

	Lateral						**Longitudinal**						**Yaw**				
	mean↓	median↓	0.25m↑	0.5m↑	1m↑	2m↑	mean↓	median↓	0.25m↑	0.5m↑	1m↑	2m↑	mean↓	median↓	1° ↑	2° ↑	4° ↑
⋆ SIBCL[33]	1.02	0.54	25.59	46.26	72.63	89.78	1.69	0.64	21.91	41.22	64.47	80.37	**1.91**	**0.85**	**56.05**	**79.70**	**90.89**
CVML[36]	3.38	2.40	6.11	12.24	23.78	44.14	3.54	2.46	5.97	11.68	23.73	43.36	-	-	-	-	-
HighlyAcc[22]	1.24	0.83	16.51	32.05	57.65	83.11	2.44	2.01	7.14	14.11	27.41	49.94	3.23	1.82	29.83	53.41	76.51
Ours	**0.14**	**0.12**	**84.58**	**99.54**	**99.98**	**100.00**	**0.10**	**0.09**	**98.55**	**100.00**	**100.00**	**100.00**	3.57	1.78	31.18	54.13	76.00

⋆: indicates LiDAR-visual hybrid methods. ↑: larger is better. ↓: lower is better.
Our method significantly improves translation accuracy while maintaining orientation accuracy compared to SOTA visual method [22].

Table 2: Comparison on the FordAV-CVL dataset

		Lateral						**Longitudinal**						**Yaw**				
		mean↓	median↓	0.25m↑	0.5m↑	1m↑	2m↑	mean↓	median↓	0.25m↑	0.5m↑	1m↑	2m↑	mean↓	median↓	1° ↑	2° ↑	4° ↑
Log4	⋆ SIBCL[33]	1.29	0.55	24.83	45.90	74.06	89.14	2.31	0.78	18.72	34.11	58.26	75.44	2.23	0.57	66.76	81.78	90.50
	CVML[36]	2.78	2.22	5.91	11.78	23.27	45.06	3.24	2.66	6.07	11.45	21.22	38.82	-	-	-	-	-
	HighlyAcc[22]	1.21	0.84	16.56	31.31	57.64	85.45	2.47	1.82	7.11	13.87	28.53	53.64	2.94	1.83	30.74	53.08	78.40
	Ours (Front)	0.94	0.54	26.11	46.73	73.48	89.69	1.56	0.80	17.95	34.44	56.30	75.41	2.77	1.18	44.47	66.61	83.26
	Ours (2FR)	0.60	0.51	24.66	48.78	82.40	98.20	0.99	0.65	22.45	41.31	64.49	86.69	1.14	0.77	60.78	85.28	96.28
	Ours (2Sides)	0.78	0.55	24.75	46.36	75.01	94.91	1.58	0.92	14.68	28.77	52.68	73.52	3.56	2.14	24.94	47.39,	71.50
	Ours (4Cams)	**0.58**	**0.46**	**26.45**	**53.60**	**85.00**	**98.81**	**0.88**	**0.49**	**25.77**	**50.31**	**75.44**	**91.53**	**0.74**	**0.50**	**77.61**	**94.94**	**98.57**
Log4→5	⋆ SIBCL[33]	1.99	1.38	10.49	21.57	39.05	64.98	6.27	3.23	13.77	22.22	31.11	42.62	3.32	1.78	31.78	54.91	78.90
	CVML[36]	3.10	2.31	5.25	10.88	20.58	43.25	3.32	2.63	5.89	10.45	21.86	39.87	-	-	-	-	-
	HighlyAcc[22]	1.69	1.61	9.45	18.03	31.72	66.06	2.99	2.32	4.63	9.69	19.89	39.28	3.35	2.44	22.43	42.32	75.19
	Ours (4Cams)	**0.96**	**0.68**	**20.03**	**37.83**	**65.09**	**87.53**	**1.43**	**0.82**	**17.45**	**33.91**	**56.73**	**76.96**	**2.76**	**1.38**	**39.20**	**61.90**	**79.37**

Log4: Localization on the same road with the different time and seasons w.r.t. the training dataset.
Log4→5: Localization on a totally different road w.r.t. the training dataset to evaluate the generalization ability.
We evaluate the effect of camera configuration on localization accuracy using multiple camera settings in Log4 (row 4-7).

methods, with a significant reduction in the translation error. Specifically, our method achieves a reduction of 86% and 94% in mean lateral and longitudinal localization error. While our orientation accuracy is slightly less accurate than the LiDAR-based method, it maintains a comparable performance to SOTA visual-only method [22] in terms of rotation error. These results demonstrate the ability of our method to generalize to a wide range of scenes.

The performance of our method on cross-season generalization is presented in 'Log4'[5] of Tab. 2. The test set in this case includes data from different time and seasons compared to the training set, which allows us to evaluate the performance of our method under varying lighting and seasonal conditions. Furthermore, in 'Log4→5' of Tab. 2, we analyze our method's generalization capability on an unseen route. In both cases, our method outperforms existing SOTA methods by significant margins. Specifically, we achieve a reduction of 52% and 43% in mean localization lateral error, 62% and 52% in mean localization longitudinal error, and 67% and 17% in mean orientation error in terms of seen and unseen routes, respectively. These results once again demonstrate the strong performance and robust generalization capabilities of our proposed method.

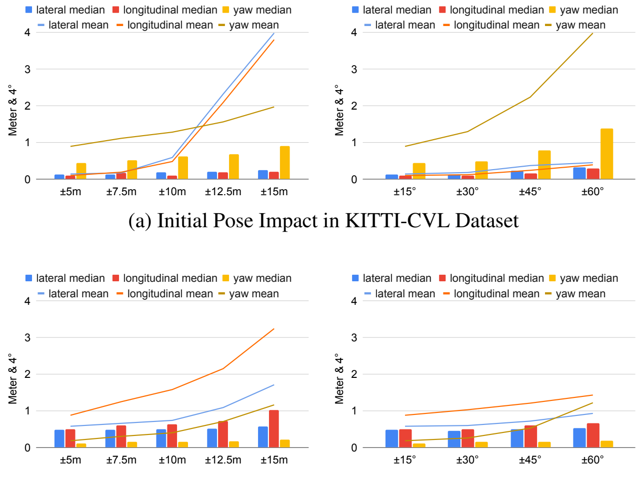
Performance with Varying Numbers of Camera Inputs. We investigate the impact of multiple onboard cameras on the FordAV-CVL dataset and evaluate our method using different camera setups. These setups include the front camera (Front) in the 1-camera setting, two side cameras (2Sides), the front and rear cameras (2FR) in the 2-camera setting, and all front, rear, and two side cameras (4Cams) in the 4-camera setting. Our findings indicate that even with the use of a single front camera ('Ours (Front)' in Tab. 2), our method outperforms the SOTA methods. Additional camera inputs lead to further improvements in performance, particularly with regards to orientation estimation, which can be attributed to the fact that a larger field of view (FoV) provides more information to accurately estimate orientation. Furthermore, our study reveals that the front and rear cameras provide more information for localization, whereas the left and right cameras contribute more to the lateral estimation. This could be attributed to the limited visibility of noticeable localization features such as road marks in the side cameras or the sensitivity of the side cameras to the roll angle. It is noteworthy that our method, despite utilizing four onboard cameras, consumes less memory (4499 MB) than HighlyAccurate [22], which requires 6445 MB due to its use of sparse purification.

Performance under Different Initial Poses. The proposed method utilizes the LM algorithm and is subject to a convergence range [6] constraint. If the provided initial pose falls outside of this range, the method may fail to converge. To evaluate the method's robustness under a more stringent

[5] The trajectory of 'Log4' was selected for method evaluation in SIBCL [33] due to its relatively good satellite view alignment. Additionally, we evaluated other logs and the evaluation results can be found in the supplementary material.

[6] The convergence range refers to the region in the pose space where the method can converge to the ground truth pose.

scenario, we conducted experiments using a comprehensive set of initial poses. The results, shown in Fig. 7, indicate that our approach achieves a satisfactory level of accuracy even when the initial pose is subjected to yaw angle errors of up to ±60° and lateral and longitudinal shifts of up to ±15m. The longitudinal estimation is found to be more sensitive to the initial pose compared to the lateral estimation. Moreover, in KITTI-CVL datasets that rely solely on a front onboard camera, a larger difference between the mean and median values suggests more cases falling outside the convergence range. Therefore, the use of multiple camera inputs, such as in the FordAV-CVL dataset with four cameras, can significantly expand both the translation and orientation coverage ranges, with the orientation coverage range being notably more improved.



(a) Initial Pose Impact in KITTI-CVL Dataset

(b) Initial Pose Impact in FordAV-CVL Dataset

Figure 7: (left) Method performance as initial pose translation varies, with orientation noise fixed within ±15° range. (right) Method performance as initial pose orientation varies, with translation noise fixed within ±5m range. The vertical axis shows translation error in units of m and orientation error in units of 4°. Additional metric results can be found in the supplementary.

6. Ablation Study

Two Confidence Maps. The proposed method adopts two types of confidence maps ("2c w/o SE"), i.e., view-consistent and on-ground maps. An alternative approach was to use a single confidence map ("1c w/o SE"), which combined both on-ground and view-consistent confidences, and disabled gradient backpropagation from the satellite view. A comparison of using different types of confidence maps is reported in Tab. 3. We can see that using two confidence maps with distinct gradient backpropagation mechanisms leads to better performance compared to the alternative approach.

Spatial Embedding. We study the impact of Spatial Embedding by comparing the performance of our algorithm with ("Full") and without Spatial Embedding ("2c w/o SE"), as shown in Tab. 3. The results demonstrate that incorporating Spatial Embedding significantly improves the performance of the PureACL algorithm.

View-consistent On-ground Keypoint Detector. We compare our keypoint detection design with the SOTA SuperPoint [3]. In this comparison, we use SuperPoint to detect keypoints and combine it with the two confidence maps to reduce the weights of points located on dynamic objects or above the ground plane. The results are presented in Tab. 3 as "SuperPoint". Our view-consistent on-ground point detector ("Full") outperforms "SuperPoint" as it detects a sufficient number of on-ground keypoints, which is more beneficial for cross-view localization.

Table 3: Ablation study on FordAV-CVL dataset

		Lateral		**Longitudinal**		**Yaw**	
FordAV-CVL		mean↓	median↓	mean↓	median↓	mean↓	median↓
1c	**w/o SE**	0.63	0.49	1.29	0.68	2.08	1.28
2c	**w/o SE**	0.63	0.48	1.17	0.63	0.90	0.57
	Full	**0.58**	**0.46**	**0.88**	**0.49**	**0.74**	**0.50**
SuperPoint[3]		0.65	0.53	0.95	0.52	1.05	0.60
Mean fusion		0.61	0.47	0.90	0.50	0.90	0.55

Our **Full** solution incorporates 2 confidence maps (2c) along with Spatial Embedding (w/ SE).

Multi-camera Fusion Method. We compare two fusion methods for keypoints captured by multiple onboard cameras: selecting the highest-confidence 2D projection ("Full"), which is used in our proposed method, and computing the mean of features and confidence scores across all visible onboard camera images ("Mean fusion"). The results in Tab. 3 show that highest-confidence fusion outperforms Mean fusion due to more reliable selection.

7. Conclusion

This paper presents PureACL, a novel cross-view localization approach for accurate 3-DoF pose estimation that supports flexible multi-camera inputs. Our approach utilizes a view-consistent on-ground keypoint detector to handle dynamic objects and viewpoint variations while removing off-the-ground objects to establish the homography transformer assumption. Additionally, PureACL incorporates a spatial embedding that maximizes the use of camera intrinsic and extrinsic information to reduce visual matching ambiguity. PureACL is the first sparse visual-only approach and the first visual-only cross-view method capable of achieving a mean translation error of less than one meter. Our future plan is to integrate PureACL into the SLAM system for reduced loop closure dependence. Ultimately, PureACL has the potential to lead to robust, reliable, accurate, and low-cost localization systems.

8. Acknowledgements

The research is funded in part by an ARC Discovery Grant (grant ID: DP220100800) to HL.

References

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

[2] Ioan Andrei Barsan, Shenlong Wang, Andrei Pokrovsky, and Raquel Urtasun. Learning to localize using a lidar intensity map. *arXiv preprint arXiv:2012.10902*, 2020.

[3] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. *CoRR*, abs/1712.07629, 2017.

[4] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017.

[5] Yanming Feng, Jinling Wang, et al. Gps rtk performance characteristics and analysis. *Positioning*, 1(13), 2008.

[6] Florian Fervers, Sebastian Bullinger, Christoph Bodensteiner, Michael Arens, and Rainer Stiefelhagen. Continuous self-localization on aerial images using visual and lidar sensors. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7028–7035. IEEE, 2022.

[7] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[8] Google. Maps static api, 2023. 2023.

[9] Frank R Hampel, Elvezio M Ronchetti, Peter J Rousseeuw, and Werner A Stahel. *Robust statistics: the approach based on influence functions*, volume 196. John Wiley & Sons, 2011.

[10] Sixing Hu, Mengdan Feng, Rang MH Nguyen, and Gim Hee Lee. Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7258–7267, 2018.

[11] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Dense visual slam for rgb-d cameras. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2100–2106. IEEE, 2013.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] Richard B Langley. Rtk gps. *Gps World*, 9(9):70–76, 1998.

[14] Liu Liu and Hongdong Li. Lending orientation to neural networks for cross-view geo-localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[15] Liu Liu, Hongdong Li, and Yuchao Dai. Efficient global 2d-3d matching for camera localization in a large-scale 3d map. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2372–2381, 2017.

[16] Ian D Miller, Anthony Cowley, Ravi Konkimalla, Shreyas S Shivakumar, Ty Nguyen, Trey Smith, Camillo Jose Taylor, and Vijay Kumar. Any way you look at it: Semantic crossview localization and mapping with lidar. *IEEE Robotics and Automation Letters*, 6(2):2397–2404, 2021.

[17] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017.

[18] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. Ieee, 2004.

[19] Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, Hao Li, and Rong Jin. Softtriple loss: Deep metric learning without triplet sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6450–6458, 2019.

[20] Tyler G.R. Reid, Sarah E. Houts, Robert Cammarata, Graham Mills, Siddharth Agarwal, Ankit Vora, and Gaurav Pandey. Localization requirements for autonomous vehicles. *SAE International Journal of Connected and Automated Vehicles*, 2(3), sep 2019.

[21] Paul-Edouard Sarlin, Ajaykumar Unagar, Måns Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, and Torsten Sattler. Back to the Feature: Learning robust camera localization from pixels to pose. In *CVPR*, 2021.

[22] Yujiao Shi and Hongdong 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.

[23] Yujiao Shi, Xin Yu, Dylan Campbell, and Hongdong 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*, pages 4064–4072, 2020.

[24] Yujiao Shi, Xin Yu, Liu Liu, Tong Zhang, and Hongdong Li. Optimal feature transport for cross-view image geo-localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11990–11997, 2020.

[25] Young-Sik Shin, Yeong Sang Park, and Ayoung Kim. Dvl-slam: Sparse depth enhanced direct visual-lidar slam. *Autonomous Robots*, 44(2):115–130, 2020.

[26] Tim Yuqing Tang, Daniele De Martini, Dan Barnes, and Paul Newman. Rsl-net: Localising in satellite images from a radar on the ground. *IEEE Robotics and Automation Letters*, 5(2):1087–1094, 2020.

[27] Tim Y Tang, Daniele De Martini, and Paul Newman. Get to the point: Learning lidar place recognition and metric localisation using overhead imagery. *Proceedings of Robotics: Science and Systems, 2021*, 2021.

[28] Tim Y Tang, Daniele De Martini, Shangzhe Wu, and Paul Newman. Self-supervised learning for using overhead imagery as maps in outdoor range sensor localization. *The International Journal of Robotics Research*, 40(12-14):1488–1509, 2021.

[29] Aysim Toker, Qunjie Zhou, Maxim Maximov, and Laura Leal-Taixé. Coming down to earth: Satellite-to-street

view synthesis for geo-localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6488–6497, 2021.

[30] Frank Van Diggelen and Per Enge. The world's first gps mooc and worldwide laboratory using smartphones. In *Proceedings of the 28th international technical meeting of the satellite division of the institute of navigation (ION GNSS+ 2015)*, pages 361–369, 2015.

[31] Lukas Von Stumberg, Patrick Wenzel, Nan Yang, and Daniel Cremers. Lm-reloc: Levenberg-marquardt based direct visual relocalization. In *2020 International Conference on 3D Vision (3DV)*, pages 968–977. IEEE, 2020.

[32] Ankit Vora, Siddharth Agarwal, Gaurav Pandey, and James McBride. Aerial imagery based lidar localization for autonomous vehicles, 2020.

[33] Shan Wang, Yanhao Zhang, Ankit Vora, Akhil Perincherry, and Hengdong Li. Satellite image based cross-view localization for autonomous vehicle. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3592–3599. IEEE, 2023.

[34] Xinkai Wei, Ioan Andrei Bârsan, Shenlong Wang, Julieta Martinez, and Raquel Urtasun. Learning to localize through compressed binary maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10316–10324, 2019.

[35] Ryan W. Wolcott and Ryan M. Eustice. Visual localization within lidar maps for automated urban driving. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 176–183, 2014.

[36] Zimin Xia, Olaf Booij, Marco Manfredi, and Julian FP Kooij. Visual cross-view metric localization with dense uncertainty estimates. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, pages 90–106. Springer, 2022.

[37] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2, pages 1–9. Berkeley, CA, 2014.

[38] 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.