

Supplemental Material: Uncertainty-aware Vision-based Metric Cross-view Geolocalization

Florian Fervers¹ Sebastian Bullinger¹ Christoph Bodensteiner¹ Michael Arens¹ Rainer Stiefelhagen²

¹Fraunhofer IOSB ²Karlsruhe Institute of Technology

¹{firstname.lastname}@iosb.fraunhofer.de ²rainer.stiefelhagen@kit.edu

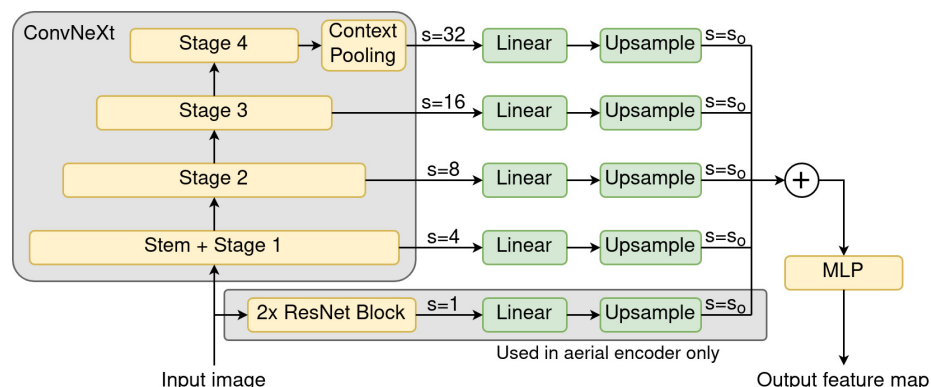


Figure 1. Flowchart of the encoder network used to predict feature maps for all input images. The model uses a pretrained vision backbone (*i.e.* ConvNeXt [25]) to extract intermediate feature maps, utilizes a global average for context pooling in the last feature map [12, 55] and predicts a final feature map at output stride s_o . In the aerial encoder we additionally process the input image using two ResNet blocks [16] at stride 1 to fully exploit the image’s spatial information.

A. Details of encoder network

Our method is agnostic to the choice of the encoder network used to extract feature maps from the input images. We design a lightweight model for this purpose as follows (*cf.* Fig. 1). We further publish the source code of the model online.

A pretrained vision backbone (*i.e.* ConvNeXt [25]) is used to extract intermediate feature maps at strides 4, 8, 16 and 32. For context pooling, we compute the global spatial average of the last feature map, process it via a small MLP and concatenate the result with the feature map along the channel dimension [12, 55]. All intermediate feature maps are mapped onto c channels via linear layers, upsampled to the desired output stride s_o and summed. We use $c = 64$ for the ablation studies and $c = 128$ for all other evaluations. The result is processed by a small pixelwise MLP to predict the final feature map.

We extract features at stride $s_o = s_A = 1$ for aerial images and $s_o = s_G = 4$ for ground images. Since the spatial resolution of the aerial feature map is particularly important for the localization accuracy, we additionally process the input image via two ResNet blocks [16] at stride 1 and include

the result in the list of intermediate feature maps.

B. Dataset overview

Table 1 provides an overview of the datasets used for the evaluation of our method. The data is captured over nine regions which allows creating disjoint train and test splits for cross-area evaluation. Ford AV and KITTI-360 contain scenes that are significantly longer than the other datasets in Table 1 and are therefore most suited for evaluating tracking frameworks.

In addition to aerial images from DCGIS, MassGIS and Stratmap (which are taken between 2017 and 2021) we collect aerial images from Google Maps and Bing Maps during 2022. These images might be several years old since the recording date is not provided.

C. Detailed results on Ford AV dataset

For the comparison with related works in Sec. 5, we evaluate our method on a test split of Ford AV according to the definition of Shi *et al.* [35]. Since their evaluation protocol considers only a subset of the first two scenes, we provide detailed results on all six trajectories (2017-08-04 V2 Log1

Table 1. Datasets used to evaluate the proposed CVGL approach. Each ground data-frame consists of the vehicle’s pose, camera images as well as intrinsic and extrinsic parameters. Data-frames are divided into disjoint cells with size $100\text{m} \times 100\text{m}$ to measure aerial coverage. SD: Average scene duration in seconds.

Dataset	Region	Year	Scenes	Frames ($\times 10^3$)	SD (sec)	Cams	Cells	Orthophoto providers
Argoverse V1 [11]	Miami	≤ 2019	53	12	22	9	71	Google Maps [3], Bing Maps [1]
	Pittsburgh	≤ 2019	60	10	17	9	55	Google Maps [3], Bing Maps [1]
Argoverse V2 [45]	Austin	≤ 2021	111	48	43	7	296	Google Maps [3], Bing Maps [1], Stratmap [5]
	Detroit	≤ 2021	256	91	36	7	569	Google Maps [3], Bing Maps [1]
	Miami	≤ 2021	703	245	34	7	811	Google Maps [3], Bing Maps [1]
	Palo Alto	≤ 2021	43	136	34	7	157	Google Maps [3], Bing Maps [1]
	Pittsburgh	≤ 2021	668	228	34	7	557	Google Maps [3], Bing Maps [1]
	Washington	≤ 2021	262	90	34	7	553	Google Maps [3], Bing Maps [1], DCGIS [2]
	Detroit	2017	18	136	811	6-7	983	Google Maps [3], Bing Maps [1]
KITTI-360 [21]	Karlsruhe	2013	9	76	877	3	609	Google Maps [3], Bing Maps [1]
Lyft L5 [18]	Palo Alto	2019	398	50	25	6	88	Google Maps [3], Bing Maps [1]
Nuscenes [9]	Boston	2018	467	19	20	6	174	Google Maps [3], Bing Maps [1], MassGIS [4]
Pandaset [49]	Palo Alto	2019	35	3	8	6	87	Google Maps [3], Bing Maps [1]
	San Francisco	2019	65	5	8	6	93	Google Maps [3], Bing Maps [1]

to 2017-08-04 V2 Log6) of the dataset w.r.t. our pseudo-labeled ground truth in Table 2.

We train our model in a cross-area setting on Argoverse V1, Argoverse V2, Lyft L5, Nuscenes and Pandaset with aerial images from Google Maps, but remove data from Detroit where Ford AV was recorded. We evaluate with a search radius of 50m and an angle noise of 30° and 360° to simulate known and unknown orientation, respectively. We provide the position and bearing offsets used for the evaluation with our code.

The recall on the first two scenes is lower than the results on the test split of Shi *et al.* [35] due to a larger search region and potentially due to the additional data-frames. While our model achieves the highest recall on the last four scenes, Shi *et al.* [35] report their lowest recall on the last four scenes in the supplementary material - likely due to worse ground truth.

D. Tracking videos

The supplementary material contains two videos of the tracking framework (*cf.* Sec. 5.3) applied to two scenes in the Ford AV and KITTI-360 datasets - demonstrating the capabilities of our proposed approach. The videos follow the predicted vehicle pose and show

- (1) the predicted probabilities by the model before being processed by the Kalman filter,
- (2) the posterior probabilities produced by the Kalman filter, and
- (3) the projected lidar points to assess the alignment of the pose with the aerial image.

Table 2. Recall in percent on all six trajectories of the Ford AV dataset [6] (2017-08-04 V2) w.r.t. our pseudo-labeled ground truth. The prior pose is chosen with up to 50m error to the vehicle position. Rotation noise is defined below.

	Angle noise	Lateral			Longitudinal		
		1.0m	3.0m	5.0m	1.0m	3.0m	5.0m
Log1	30°	63.8	87.4	91.1	27.4	61.1	67.6
Log2	30°	58.7	83.2	85.6	21.8	53.4	60.7
Log3	30°	90.1	99.2	99.5	77.5	98.1	99.0
Log4	30°	89.9	99.7	100.0	71.6	95.8	97.8
Log5	30°	89.5	99.8	99.8	78.7	98.5	98.9
Log6	30°	87.8	97.9	98.3	69.9	94.1	95.3
Log1	360°	54.8	72.7	75.9	23.2	53.1	59.7
Log2	360°	44.1	62.8	65.4	17.4	41.3	48.0
Log3	360°	90.3	98.8	99.0	77.3	97.4	98.3
Log4	360°	89.3	98.9	99.4	70.8	94.6	97.2
Log5	360°	89.5	99.7	99.8	79.2	98.2	98.5
Log6	360°	86.6	96.1	96.7	70.2	93.5	94.8