CrossLoc: Scalable Aerial Localization Assisted by Multimodal Synthetic Data Supplementary Materials

Qi Yan, Jianhao Zheng, Simon Reding, Shanci Li, Iordan Doytchinov
École Polytechnique Fédérale de Lausanne (EPFL)
{firstname.lastname}@epfl.ch
crossloc.github.io

In the supplemental material, we provide additional details about the TOPO-DataGen workflow, the proposed benchmark datasets, and the experiments carried out in the main paper. Specifically, we discuss the following topics:

1. The error analysis of our benchmark datasets and the overall potential societal impact.
2. The implementation details of the proposed CrossLoc and the adapted DDLoc baseline.
3. Additional qualitative comparisons and discussion on the failure cases and limitations.

All the source code, the proposed benchmark datasets and the pre-trained model weights are available at crossloc.github.io .

1. Benchmark datasets error analysis

We validate the accuracy of the multimodal synthetic data generated through our TOPO-DataGen and the geo-tag quality of the real images captured by our drone [1] equipped with the real-time kinematics (RTK) level of Global Navigation Satellite System (GNSS) positioning.

1.1. Synthetic data quality control

Geo-referenced 3D scene model precision. For the whole benchmark datasets, the digital surface models (DTM) [5] and the LiDAR point clouds [4] used for rendering were acquired with planimetric precision of $\pm$ 20 cm and altimetric precision of $\pm$ 10 cm. We employ the orthophoto assets with a position accuracy of $\pm$ 15 cm and a resolution of 10 cm [3] to colorize the DTM and point clouds. The $\pm$ sign denotes the standard deviation w.r.t. the local coordinate reference systems [2]. In the pre-processing step, we convert the open geodata into the global WGS84 coordinate reference system, and the loss of accuracy therein is negligible. Please see our source code for more details.

Scene coordinate ray tracing accuracy. The ray-traced scene coordinate label accuracy is further evaluated by verifying the camera pose computation. Table 1 demonstrates how well the generated coordinate maps correspond to the ground truth virtual camera viewpoints. The mean camera pose computation errors for the Urbanscape and Naturescape datasets are 0.11m, 0.06°, and 0.23m, 0.06°, respectively. We use the eight-times-downsampled scene coordinates in this evaluation, the same dataset used in the main paper's experiments.

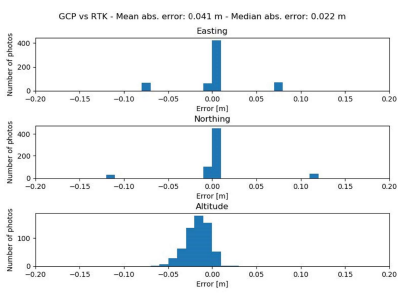
	Urbanscape		Naturescape	
	transl.	rot.	transl.	rot.
Mean	0.11m	0.06°	0.23m	0.06°
Std.	0.06m	0.03°	0.09m	0.03°
Median	0.10m	0.06°	0.22m	0.06°

Table 1: Indirect quality estimation of the scene coordinate labels. We feed the generated coordinate maps into the DSAC* [8] PnP solver and compute the poses error w.r.t. the ground truth camera viewpoints.

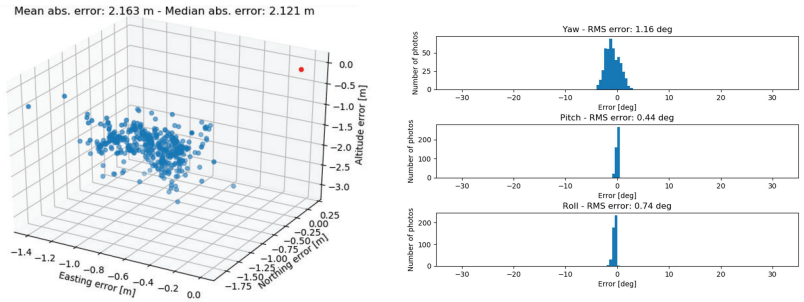
	Urbanscape	Naturescape
Mean	1.19px	1.04px
Std.	0.16px	0.11px
Median	1.14px	1.03px
90%	1.44px	1.08px
95%	1.52px	1.09px
99%	1.68px	1.14px

Table 2: Absolute reprojection error of the scene coordinate labels. Following [12], we report the reprojection error in the image plane in pixels and particularly show the percentile errors to indicate the long tail distribution.

Table 2 reports the reprojection error of the scene coordinates w.r.t. to the ground-truth camera viewpoints as in [12]. The proposed benchmark datasets have consistently low error at the magnitude of one or two pixels in both scenes. The percentile errors are close to the mean or median errors and indicate that there are few occurrences far away from the central distribution. The full-size coordinate maps (without downsampling) are used in the reprojection error analysis.



(a) Urbanscape RTK precision: position error.



(b) Naturescape RTK precision. **Left.** Position error. **Right.** Orientation error.

Figure 1: Accuracy evaluation of the real data RTK geo-tag. We show the camera pose refinement error obtained by solving the photogrammetry bundle adjustment via GCP alignment. The RTK precision in the Urbanscape dataset is at cm level, while it degrades to over 2 meters in the Naturescape dataset because of weak GNSS signals.

1.2. Real data quality control

To collect high-accuracy aerial photos, a DJI Phantom 4 RTK drone [1] was used, whose RTK positioning enables a cm-level accuracy for the image geo-tags. A network of ground control points (GCP) was measured with a JAVAD Triumph-LS [6] base with mm-level accuracy. We use 13 and 6 GCPs, respectively, for the Urbanscape and the Naturescape datasets to solve the bundle adjustment photogrammetric alignment and compare with the RTK geo-tags extracted from the photo metadata. Figure 1 shows the RTK geo-tag error statistics w.r.t. the computed photogrammetry ground truth. The geo-tag precision at the Urbanscape dataset is as small as 4 cm, while it degrades significantly in the Naturescape dataset.One can observe that there is a mean positional error bias. In the next steps, this bias was subtracted for all images of each drone, bringing the error to a lower level (mean error = 0.42 m).

2. Potential societal impact

TOPO-DataGen and benchmark datasets. The proposed TOPO-DataGen is a multi-purpose task-agnostic synthetic data generation tool that entails little ethical concerns. Essentially, it needs real-world geo-data to perform data rendering, most of which is provided by national agencies [9]. The incoming researchers adopting our method are advised to pay attention to the privacy or transparency of the underlying geo-data before implementation. In our benchmark datasets, the open-source swisstopo [3, 4, 5] geo-data is employed for synthetic data generation, and the real data is collected using a drone. The produced images and associated labels have airborne perspectives, distinct from those in indoor or urban street scenes. There is no human object data or personally identifiable information in our datasets.

CrossLoc localization. Our CrossLoc is a scene coordinate regression-based localization method and does not impose particular requirements infringing privacy. It learns to localize the image primarily based on the distinct geometries in the environment, such as mountains or buildings. It does not require human data and is unlikely to benefit from any.

3. Implementation details

3.1. CrossLoc architecture

Network architecture. Following [8], we adopt a fully convolutional network with ResNet-style skip layers [11] to employ the diverse data augmentation, including rescaling and rotation. Table 4 shows the CrossLoc encoder-decoder network and the projection head for representations concatenation. We apply group normalization [20] and `relu` nonlinear activation function for each convolutional or residual layer. The parameter N in Table 4b is task dependent, *e.g.*, for coordinate regression $N = 4$ because of 3-dimensional coordinate prediction and 1-dimensional uncertainty estimation. The parameter T in Table 4c refers to the number of visual representations; for the vanilla *CrossLoc*, $T = 3$, and for the *CrossLoc-SE* using external semantics, $T = 4$. The sign $+$ and $\bigoplus$ respectively stand for addition and channel-wise concatenation operators. As in [8], we use consecutive three convolutional layers with strides of two to downsample the prediction eight times. For the semantic segmentation task, we keep the full-size labels and apply the dense upsampling convolution [19] in the final output layer to recover the full-size semantic prediction.

Training hyper-parameters. In the first step of CrossLoc training, we pretrain the sub-task networks with task-agnostic *LHS-sim* synthetic data, and fine-tune the models with pairwise real-sim data. Table 3 shows the specific training hyper-parameters. The learning rate is halved at 50 and 100 epochs at most twice. We extend the encoder fine-tuning epochs particularly on the out-of-place scene in both datasets to ensure convergence. Notably, the semantic

Task	Urbanscape						Naturescape					
	Encoder pretraining		Encoder finetuning with in-place data		Encoder finetuning with out-of-place data		Encoder pretraining		Encoder finetuning with in-place data		Encoder finetuning with out-of-place data	
	Epoch	Initial LR	Epoch	Initial LR	Epoch	Initial LR	Epoch	Initial LR	Epoch	Initial LR	Epoch	Initial LR
Coordinate	150	0.0002	150	0.0002	1500	0.0002	100	0.0002	150	0.0002	2000	0.0002
Depth	150	0.0002	150	0.0002	300	0.0002	100	0.0002	150	0.0002	2000	0.0002
Surface normal	150	0.0002	150	0.0002	300	0.0002	100	0.0002	150	0.0002	2000	0.0002
Semantics	30	0.0002	30	0.0002	30	0.0002	30	0.0002	30	0.0002	30	0.0002

Table 3: CrossLoc encoder-decoder initialization training hyper-parameters.

segmentation tasks reach convergence much faster than the other regression tasks. Subsequently, during coordinate network fine-tuning using the frozen non-coordinate encoders as feature extractors, we train each model for 1000 epochs with a fixed learning rate of 0.0001. In line with [17, 21], we find that reusing fixed modules instead of training from scratch simultaneously improves training convergence. The Adam optimizer [13] is used throughout our training.

3.2. DDLoc architecture

ARC structure. DDLoc is our adaption of the attend-remove-complete (ARC) framework [21], which is a state-of-the-art domain transformation method. The original ARC architecture contains a style translator mapping images between real-world and synthetic domains. It further trains an attention module to detect challenging regions and an inpainting module to complete the masked regions with realistic fill-in. A depth predictor module then takes the translated result as input to make the prediction. The ARC method has been verified by extensive experiments that it can leverage synthetic data for accurate depth estimations [21].

Network architecture. In our implementation of DDLoc, the depth predictor is replaced with a coordinate regressor, which is implemented by an encoder-decoder architecture [23] with skip connections [22]. Based on that, the scene coordinate prediction is further down-sampled by the factor of 8 to enhance efficiency as well as to increase the receptive field [8]. The down-sampling is implemented by fully convolution layers with stride of 2. Any other single module in DDLoc uses the same encoder-decoder architecture as used in [23]. The decoder of the attention module is modified to output a single channel to discover challenging regions from real-world input images.

Training hyper-parameters. Following ARC's training method [21], we first pretrain each module individually using the Adam optimizer [13] with an initial learning rate of 1e-4 and coefficients of 0.9 and 0.999. For training the coordinate regressor, we adapt the original depth loss for the coordinate regression distance, and employ re-projection loss as in DSAC* [8]. The sparsity level ρ is chosen as 0.9 when training the attention module on the Urbanscape and the Naturescape datasets. Other modules are trained with the same loss as that in the original ARC implementation. Lastly, we fine-tune the whole framework with the loss of the coordinate predictor pretraining and the same training parameters.

4. Additional qualitative results and analysis

In this section, we visualize additional comparisons of the scene coordinate regression errors in Figure 2 and compare the predicted point clouds of several urban and natural scenarios in Figure 3. We show examples where our CrossLoc outputs accurate coordinate prediction and failure cases where it makes much worse estimation. Eventually, we summarize the technical limitations of our proposed methods.

Failure cases. Generally, CrossLoc and its variants outperform the others by a clear margin. Nevertheless, there are two exceptional cases where the CrossLoc family cannot make a good prediction. First, novel objects, such as construction cranes, are challenging to the CrossLoc. We conjecture that CrossLoc learns the geometric information of the buildings as a whole. Thus, the appearance of novel objects makes the scene less recognizable for CrossLoc and leads to exacerbated predictions. Second, CrossLoc is less robust to the change of illumination conditions. The error of CrossLoc prediction increases significantly in light regions and dark shadows in the Naturescape dataset. On the contrary, DDLoc predicts the coordinates in these regions more accurately thanks to the translator and attention module.

Point cloud visualization. As can be observed in Figure 3, CrossLoc gives a complete reconstruction of buildings in the Urbanscape dataset with the fewest outliers. The predicted point clouds in the Naturescape dataset are generally noisier, which is in line with the quantitative results in the paper. However, one can still observe that the points predicted by our CrossLoc are less deviated from their actual positions than the other two baselines.

Technical limitations. Firstly, the proposed TOPO-DataGen toolkit requires sufficiently-accurate geo-data for training data generation. Although there are more and more sources of open geo-data nowadays from the national agencies [9], it is not likely that they are easily accessible in

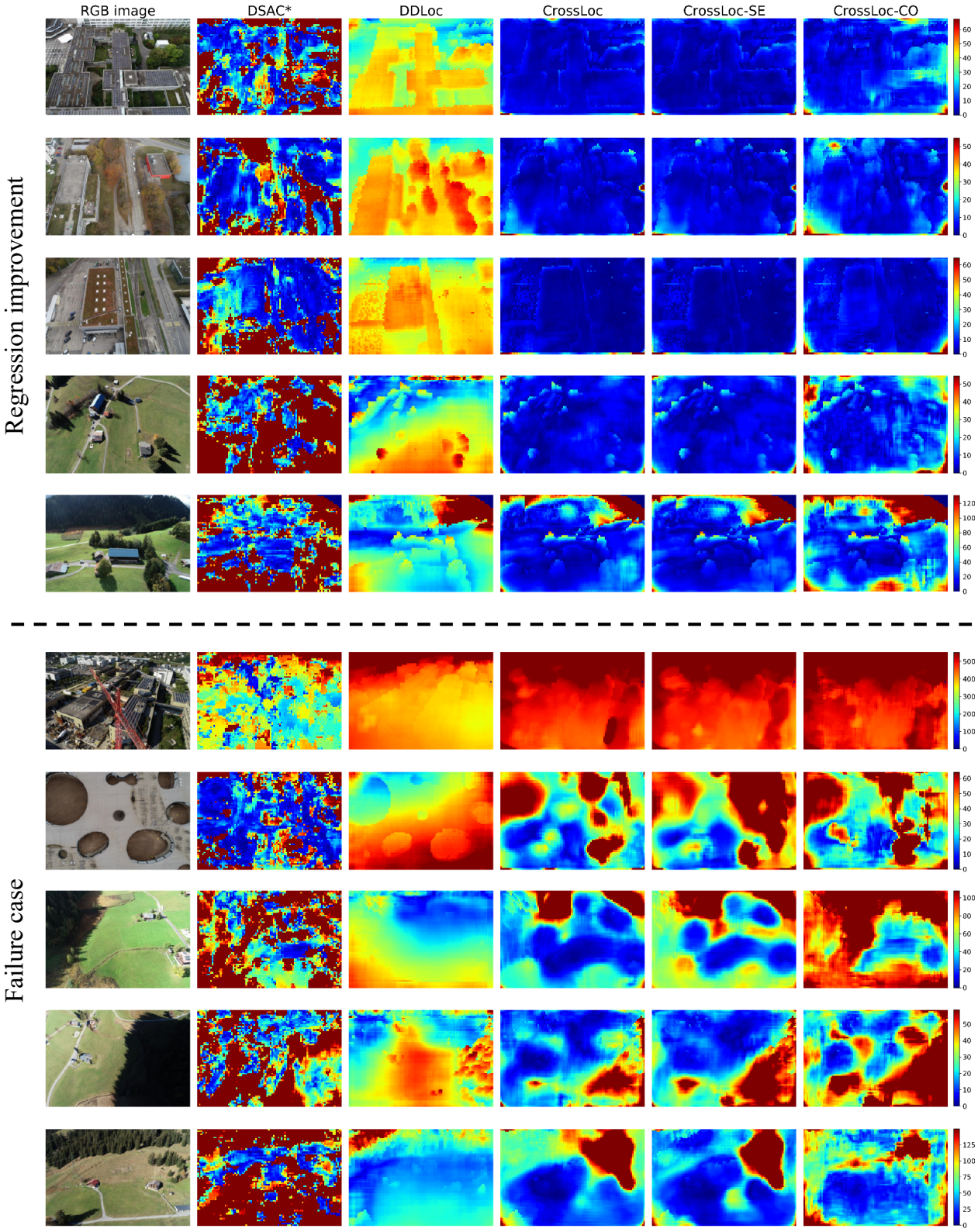


Figure 2: Additional qualitative comparison of the scene coordinate error map including improvement and failure cases. We use the same color bar for visualizing coordinate regression error in each row with the unit in meter.

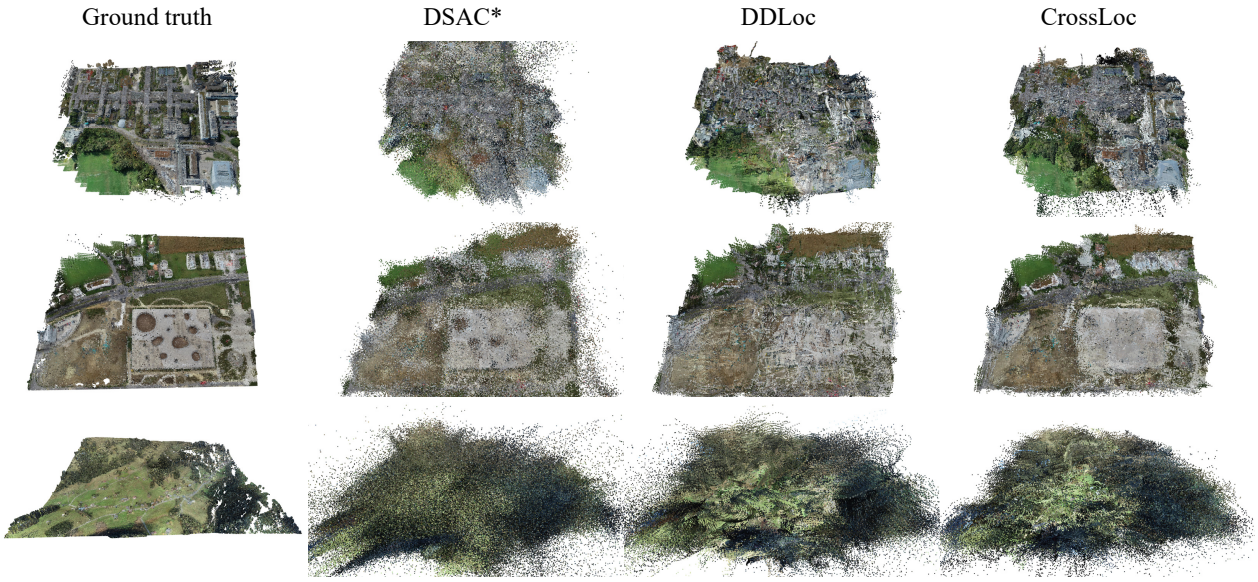


Figure 3: Comparison of point clouds predicted by different coordinate regression approaches. Our CrossLoc method generates a more complete and robust reconstruction of the buildings than the other two baselines.

any location. The data assets are critical for applying our proposed TOPO-DataGen, which is similar to many other data-driven practices.

Besides, the CrossLoc relies on CNN-extracted visual representations, and like many peer methods [8, 15], it could be specific to the local environment or texture. It is prone to failure for significant outliers or samples unforeseen during the training stage. The scalability of the proposed CrossLoc may also be limited by the capacity of the CNN backbone network, but it could be addressed by the ensemble regressor learning [7] or using more expressive backbone such as the transformers [10, 14, 18, 16].

Layer	Channel I/O	Kernel/Str./Pad.	Input
conv1	3/32	3/1/1	image
conv2	32/64	3/2/1	conv1
conv3	64/128	3/2/1	conv2
conv4	128/256	3/2/1	conv3
res1_conv1	256/256	3/1/1	conv4
res1_conv2	256/256	1/1/0	res1_conv1
res1_conv3	256/256	3/1/1	res1_conv2
res2_add	-/-	-/-/-	relu(res1_conv3+conv4)
res2_conv1	256/512	3/1/1	res2_add
res2_conv2	512/512	1/1/0	res2_conv1
res2_conv3	512/512	3/1/1	res2_conv2
res2_conv_s	256/512	1/1/0	res2_add
res3_add	-/-	-/-/-	relu(res2_conv3+res2_conv_s)
res3_conv1	512/512	3/1/1	res3_add
res3_conv2	512/512	1/1/0	res3_conv1
res3_conv3	512/512	3/1/1	res3_conv2
res4_add	-/-	-/-/-	relu(res3_conv3+res3_add)
res4_conv1	512/512	3/1/1	res4_add
res4_conv2	512/512	1/1/0	res4_conv1
res4_conv3	512/512	3/1/1	res4_conv2
feat_enc	-/-	-/-/-	relu(res4_conv3+res4_add)

(a) Encoder architecture.

Layer	Channel I/O	Kernel/Str./Pad.	Input
res1_conv1	512/512	3/1/1	feat_dec
res1_conv2	512/512	1/1/0	res1_conv1
res1_conv3	512/512	3/1/1	res1_conv2
res2_add	-/-	-/-/-	relu(res1_conv3+feat_dec)
res2_conv1	512/512	3/1/1	res2_add
res2_conv2	512/512	1/1/0	res2_conv1
res2_conv3	512/512	3/1/1	res2_conv2
res3_add	-/-	-/-/-	relu(res2_conv3+res2_add)
res3_conv1	512/512	1/1/0	res3_add
res3_conv2	512/512	1/1/0	res3_conv1
res3_conv3	512/512	1/1/0	res3_conv2
fc1	512/512	1/1/0	relu(res3_conv3+res3_add)
fc2	512/512	1/1/0	fc1
output	512/N	1/1/0	fc2

(b) Decoder architecture.

Layer	Channel I/O	Kernel/Str./Pad.	Input
feat_add	-/-	-/-/-	feat_enc_1 $\bigoplus \cdots \bigoplus$ feat_enc_T
feat_conv1	512T/512	3/1/1	feat_add
feat_conv2	512/512	1/1/0	feat_conv1
feat_conv3	512/512	3/1/1	feat_conv2
feat_skip	512T/512	1/1/0	feat_add
feat_cat	-/-	-/-/-	relu(feat_conv3 + feat_skip)

(c) Representation projection head architecture.

Table 4: CrossLoc network architecture.

References

[1] DJI PHANTOM 4 RTK. https://www.dji.com/phantom-4-rtk. 1, 2

[2] Local Swiss reference frames. https://www.swisstopo.admin.ch/en/knowledge-facts/surveying-geodesy/reference-frames/local.html. 1

[3] SWISSIMAGE 10 cm. https://www.swisstopo.admin.ch/en/geodata/images/ortho/swissimage10.html. 1, 2

[4] swissSURFACE3D. https://www.swisstopo.admin.ch/en/geodata/height/surface3d.html. 1, 2

[5] swissSURFACE3D Raster. https://www.swisstopo.admin.ch/en/geodata/height/surface3d-raster.html. 1, 2

[6] TRIUMPH-LS — JAVAD GNSS. https://www.javad.com/jgnss/products/receivers/triumph-ls.html. 2

[7] E. Brachmann and C. Rother. Expert sample consensus applied to camera re-localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7525–7534, 2019. 5

[8] E. Brachmann and C. Rother. Visual camera re-localization from rgb and rgb-d images using dsac. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1, 2, 3, 5

[9] S. Coetzee, I. Ivánová, H. Mitasova, and M. A. Brovelli. Open geospatial software and data: A review of the current state and a perspective into the future. *ISPRS International Journal of Geo-Information*, 9(2):90, 2020. 2, 3

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 5

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[12] A. Jafarzadeh, M. L. Antequera, P. Gargallo, Y. Kuang, C. Toft, F. Kahl, and T. Sattler. Crowddriven: A new challenging dataset for outdoor visual localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9845–9855, 2021. 1

[13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 3

[14] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 5

[15] P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, et al. Back to the feature: Learning robust camera localization from pixels to pose. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3247–3257, 2021. 5

[16] W. Shan, Z. Liu, X. Zhang, S. Wang, S. Ma, and W. Gao. P-stmo: Pre-trained spatial temporal many-to-one model for 3d human pose estimation. *arXiv:2203.07628*, 2022. 5

[17] W. Shan, H. Lu, S. Wang, X. Zhang, and W. Gao. Improving robustness and accuracy via relative information encoding in 3d human pose estimation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3446–3454, 2021. 3

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 5

[19] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell. Understanding convolution for semantic segmentation. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 1451–1460. IEEE, 2018. 2

[20] Y. Wu and K. He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 2

[21] Y. Zhao, S. Kong, D. Shin, and C. Fowlkes. Domain decluttering: Simplifying images to mitigate synthetic-real domain shift and improve depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3330–3340, 2020. 3

[22] C. Zheng, T.-J. Cham, and J. Cai. T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018. 3

[23] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 3