Supplementary Material
CrossLocate: Cross-modal Large-scale Visual Geo-Localization in Natural Environments using Rendered Modalities

Jan Tomešek, Martin Čadík, Jan Brejcha
Brno University of Technology, Faculty of Information Technology, CPhoto@FIT
Božetěchova 2, 61200 Brno, Czech Republic
{itomesek,cadik,ibrejcha}@fit.vutbr.cz

1. CrossLocate method details

We provide additional details about our CrossLocate localization method, especially our weak localization supervision used during the training process.

1.1. Weak localization supervision

In order to form a training triplet for each query photograph, examples of positive and negative database views need to be collected. Similarly to NetVLAD [2], we make use of the known geographic position and orientation of each training image. However, only *potentially positive* images and *definitely negative* images can be selected this way, as there is no guarantee that two images really depict the same scene, even though their geographic information is similar. This is due to various obstacles in scenes, *e.g.* overhanging cliffs, or just noisy GPS information.

We work in the WGS-84 coordinate system and perform a three-step process to select the *(potentially) positive* and *(definitely) negative* examples.

When working with the *Sparse dataset*, positive examples need to be within 20 meters from a query, while negative examples need to be farther than 2000 meters. For the *Uniform dataset*, positive examples are closer than 1000 meters and negative examples are farther than 5000 meters. Positive examples are also required to have their yaw angle within a specific angle distance from the query yaw angle, while the orientation of negative examples is not constrained in any way. This yaw angle distance is 15° for the *Sparse dataset* and 30° for the *Uniform dataset*. The negative examples are sampled randomly from the databases and filtered according to the described requirements.

As the second step, the positive and negative candidates are sorted according to their visual similarity with the corresponding query, *i.e.* based on descriptor distances, and only the most similar candidates are kept. This way, we select both well-matching positive examples and difficult negative examples. This comes at the cost of regularly recomputing query and database descriptors during the training.

Finally, only *violating negatives* are kept. The violating negatives are required to have their descriptor distances to the corresponding query descriptor smaller than the distance (increased by a margin) between the query and its positive example.

We apply geometric augmentations to both query photographs and database synthetic views. Images are randomly shifted, rotated and flipped. As the flip operation would disrupt the correspondences between query and database images, it is always applied to all images in a triplet. For additional augmentation of query photographs, we modify brightness, hue, saturation and contrast, and also add blur and noise.

Each resulting triplet is then composed of 1 query (anchor) photograph, 1 positive database view and 5 negative database views. To better utilize the time spent by assembling the triplets, we reuse each triplet three times, each time with different augmentations. We also store all the found violating negatives so that they can be used in subsequent training epochs, if finding new violating negatives becomes difficult.

1.2. Architecture

A more detailed description of our method's architecture is provided in the form of Table 1. The architecture consists of 5 convolutional blocks. Each block contains 2-3 convolutional layers (3×3 kernel) with ReLU units and is ended with a max-pooling layer. At the very end of the final convolutional block, we do not use any pooling or ReLU unit to not restrict our representation. The final L2-normalizations together with a global maximum pooling are crucial to produce our global "cross-modal" representation (descriptor) for each image.

1.3. Technical details

We train for 100 epochs when using the *Sparse dataset* and for 50 epochs when using the *Uniform dataset*. When training on the *Uniform dataset*, the *uniform compact* ver-

Layer	Dimensions
Input	(500, 500, 3)
Conv 1 (with ReLU)	(500, 500, 64)
Conv 2 (with ReLU)	(500, 500, 64)
Max-pooling 1	(250, 250, 64)
Conv 3 (with ReLU)	(250, 250, 128)
Conv 4 (with ReLU)	(250, 250, 128)
Max-pooling 2	(125, 125, 128)
Conv 5 (with ReLU)	(125, 125, 256)
Conv 6 (with ReLU)	(125, 125, 256)
Conv 7 (with ReLU)	(125, 125, 256)
Max-pooling 3	(63, 63, 256)
Conv 8 (with ReLU)	(63, 63, 512)
Conv 9 (with ReLU)	(63, 63, 512)
Conv 10 (with ReLU)	(63, 63, 512)
Max-pooling 4	(32, 32, 512)
Conv 11 (with ReLU)	(32, 32, 512)
Conv 12 (with ReLU)	(32, 32, 512)
Conv 13	(32, 32, 512)
L2-norm	(32, 32, 512)
Global max-pooling	(512)
L2-norm	(512)

Table 1. CrossLocate architecture together with activation dimensions ($height \times width \times channels$).

sion of the training database is always used (161K images instead of 7.8M). Evaluation (testing) is always done on the fully-uniform "non-compact" databases (specifically their testing sets). We use the Adam optimizer with a learning rate of 0.00001. We form batches of 3 triplets and require only 8 GB of GPU memory. The query and database descriptors are recomputed every 250 triplets for the *Sparse dataset* and every 1000 triplets for the *Uniform dataset*. For triplet loss, we use a margin of 0.1.

Validation sets of the datasets are used to select the best trained models – the models with the best combined localization performance at top-1, top-10 and top-100 candidates, and at a 1 kilometer location threshold.

For testing, we mainly evaluate recall@1 and recall@100 at numerous location thresholds/errors. For the *Sparse dataset*, we measure the performance from 0 meter tolerance to 5000 meters, every 20 meters. For the *Uniform dataset*, we measure the performance from 0 meters to 10000 meters, every 100 meters.

2. CrossLocate dataset details

We provide additional details regarding our datasets, as well as the description of the process behind their creation.

2.1. Query photographs

We use two existing datasets of photographs captured in nature – the GeoPose3K dataset [3] and the Landscape AR dataset [4].

The original Landscape AR dataset consists of 16K photographs automatically collected from the Internet, with their positions and orientations estimated using Structure-from-Motion. As our goal is to reserve the area of Switzerland for the testing set, we leave out a cluster of images around the Matterhorn mountain. This cluster would extend to both training and testing set areas. This way, we ensure our sets are strongly separated. This results in 9K photographs usable as queries for our purposes. By combining these photographs with the GeoPose3K dataset, we create a "CrossLocate" query dataset of 12K photographs from across the Alps.

2.2. CrossLocate datasets

We provide the detailed numbers of images available in our *Sparse dataset* (Table 2) and *Uniform dataset* (Table 3).

For testing purposes, we also combined our *uniform database* (used mainly in the *Uniform dataset*) with other datasets of query photographs – the CH1 dataset [7] of 203 photographs and the CH2 dataset [7] of 949 photographs. For the CH1 dataset, we extracted the geographic information of 196 photographs already included in the GeoPose3K dataset [3], and we manually found the missing geographic information (incl. orientation) for the remaining 7 photographs. In the CH2 dataset, only geographic positions are available.

The set splits of all the datasets used in our work, as well as the geographic information for all the images, are available on the project webpage (`http://cphoto.fit.vutbr.cz/crosslocate/`) as part of our benchmark.

2.3. Image modalities

Both the *sparse database* and the *uniform database* are composed of three different synthetic image modalities – *semantic segmentations*, *silhouette maps* and *depth maps*. All the image modalities are obtained by rendering a publicly available geo-referenced DEM terrain model of the Alps (`http://www.viewfinderpanoramas.org`) with a resolution of approx. 30m/px.

To render the semantic segmentations, we overlay the DEM model with the most frequent physical land cover features from the OpenStreetMap (`https://www.openstreetmap.org`): forests, glaciers, water bodies, and bare rock. These features provide mainly areal information, as the segment boundaries tend to be imprecise.

The silhouette maps were obtained by ray-casting the DEM model – the silhouettes are generated at terrain discontinuities and provide more information compared to a simple horizon line.

Sparse dataset	train	val	test	total
sparse database	16 908	6 192	14 232	37 332
queries (GeoPose3K [3])	1 409	516	1 186	3 111

Table 2. Numbers of query and database images in the *Sparse dataset*, and their split into training, validation and testing sets.

Uniform dataset	train	val	test	total
uniform database	7 849 968	889 212	1 979 160	10 718 340
uniform compact database	161 292	889 212	1 979 160	3 029 664
queries ("CrossLocate")	8 324	516	3 513	12 353

Table 3. Numbers of query and database images in the *Uniform dataset*, and their split into training, validation and testing sets.

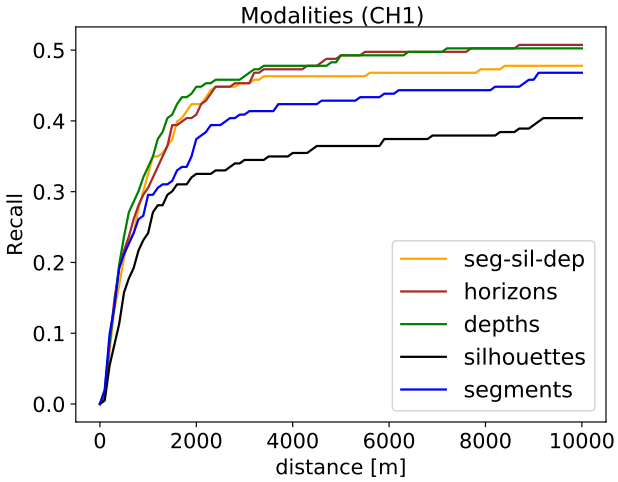

Figure 1. Localization performance of database image modalities. Results are measured on the combination of the CH1 dataset (used as queries) and the testing part of the *uniform database*.

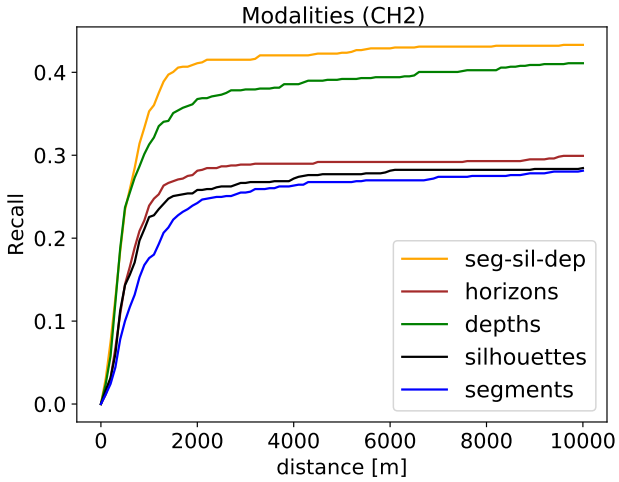

Figure 2. Localization performance of database image modalities. Results are measured on the combination of the CH2 dataset (used as queries) and the testing part of the *uniform database*.

Similarly to the silhouette maps, the depth maps were produced by ray-casting the DEM terrain model. This allowed us to compute the absolute physical distance (in meters) between the camera and the terrain at each pixel.

3. Experiments

To further support our results and claims, we extend the evaluations presented in the main text with results measured on additional testing sets.

3.1. Modalities evaluation

We provide an extended comparison of the localization performance of the individual database image modalities. The performance is measured on the combination of the testing part of the *uniform database* with the query photographs of the CH1 dataset (Fig. 1) and the CH2 dataset (Fig. 2).

The results again confirm the dominant position of depth maps among other modalities. The possible benefit of combining the base three modalities (`seg-sil-dep`) can (only) be seen on the CH2 dataset. Horizon lines generally lead to poor performance. However, the CH1 dataset is an exception to this, as it is the only set where horizon lines reach nearly the same performance as depth maps.

3.2. Comparison with the state-of-the-art methods

We provide an extended comparison of our CrossLocate approach with the HLoc method [7], DELG method [5] and HOW method [8]. The results are measured on the testing set of the *Sparse dataset* (Fig. 3), as well as on the combination of the testing part of the *uniform database* with the query photographs of the CH1 dataset (Fig. 4) and the CH2 dataset (Fig. 5).

For DELG, we provide results for the retrieval step (`DELG (global)`), as well as the geometric verification (`DELG (local)`). For HOW, we mainly provide retrieval results with global descriptors (`HOW (global)`). A result for the ASMK aggregation used in HOW (`HOW (ASMK)`) is available only on the *Sparse dataset*, because

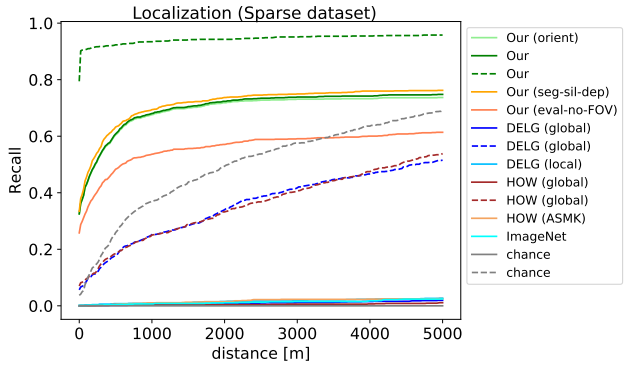

Figure 3. Localization performance of our approach and state-of-the-art methods evaluated on the testing set of the *Sparse dataset*. Solid lines: recall at 1 database candidates retrieved for each query, dashed lines: recall at 100 candidates.

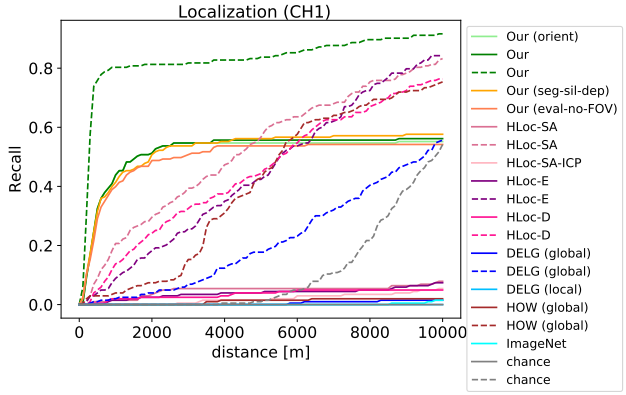

Figure 4. Localization performance of our approach and state-of-the-art methods evaluated on the combination of the CH1 dataset (used as queries) and the testing part of the *uniform database*. Solid lines: recall at 1 database candidates retrieved for each query, dashed lines: recall at 100 candidates.

of the high memory requirements related to using the *uniform database*. For HLoc, we provide results for the retrieval step performed based on (query) horizon lines automatically detected by the Edge-less method [1] (`HLoc-E`) and Deeplab method [6] (`HLoc-D`). For HLoc evaluated on the CH1 dataset, we also provide a result based on horizons lines extracted semi-automatically with human guidance [7] (`HLoc-SA`). This shows that our choice of the methods for automatic horizon detection does not have a significant impact on the performance of HLoc.

The results for recall@1 (solid lines) show that only our CrossLocate approach reaches reasonable performance (`Our`). Moreover, our recall@1 results significantly outperform the recall@100 (dashed lines) of the competing methods, up to at least 5 kilometer threshold.

For an easier comparison, only the correctness of position estimates is evaluated, without considering the correctness of orientation estimates (yaw angle). Therefore, chances of making a successful localization randomly (`chance`) are twelve times higher than when considering the correctness of orientation (because of the 12 database views at each position). This is arguably the reason why the bad performing methods DELG, HOW and HLoc are able to reach some level of performance at recall@100. However, chance plays no role in our CrossLocate approach, as the results stay nearly the same when considering the correctness of the yaw angle (30° tolerance) (`orient`). An orientation result for the CH2 dataset is not provided, because this dataset does not contain the orientation information. The result for the combination of the database modalities (`seg-sil-dep`) shows no clear benefit compared to our choice of depths maps as the main database modality. Discarding the field of view information used for the scaling of query images during evaluation leads to a small decrease in performance (`eval-no-FOV`). The importance of our cross-modal training is emphasized by the result obtained when our model is only initialized based on the ImageNet dataset, but not trained further (`ImageNet`).

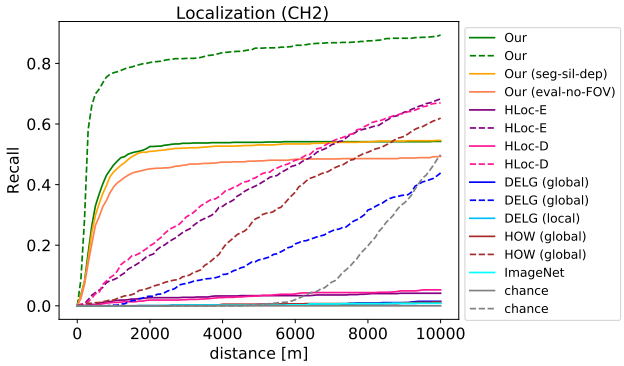

Figure 5. Localization performance of our approach and state-of-the-art methods evaluated on the combination of the CH2 dataset (used as queries) and the testing part of the *uniform database*. Solid lines: recall at 1 database candidates retrieved for each query, dashed lines: recall at 100 candidates.

Specific recall values measured on the testing set of the *Uniform dataset* can be seen in Table 4 (corresponding chart is available in the main text).

4. Qualitative evaluation

We provide a qualitative evaluation of our CrossLocate approach in the form of successful and unsuccessful localizations (Figures 6 and 7 respectively). The localization examples are taken from the evaluation conducted on the testing set of the *Uniform dataset* (3.5K query photographs and 2M database images). For each query photograph, we show the top 3 retrieved candidates, together with their distances from the query ground-truth (location distance and yaw angle distance). For the purpose of this evaluation, a localization is considered successful if the top 1 candidate

CrossLocate	recall@1			recall@100		
Uniform dataset	1 km	5 km	10 km	1 km	5 km	10 km
position	38.66	50.10	51.75	72.62	85.08	93.17
position & orientation	38.49	49.33	49.79	71.90	79.68	83.60
no FOV evaluation	31.45	45.60	47.77	65.61	81.55	91.35

Table 4. Results of our CrossLocate approach measured on the testing set of the challenging *Uniform dataset*. We provide recall values obtained when only the correctness of location estimates is considered (`position`); when also the correctness of orientation estimates is considered (`position & orientation`); and when field of view information is not utilized during the evaluation process (`no FOV evaluation`, only position is considered).

is strictly within 1 kilometer and 30° (yaw angle) from the query ground-truth (shown with a light green border). Localizations within 10 kilometers and 30° are emphasized by a dark green color. The database candidates are accompanied by scales showing depth in kilometers.

As seen in Fig. 6, our approach can deal with various obstacles and challenging weather. Specifically, our approach can successfully localize images even when horizon is hardly visible, typically because of clouds and fog. Also, examples where the scene is significantly occluded are shown. Finally, our approach works well even in difficult lighting conditions, such as in the dark.

The unsuccessful localizations show many of the difficulties related to the localization in natural environments (Fig. 7). Examples with nearly no distinctive information usable for localization are shown, as well as severely occluded scenes. Furthermore, the high self-similarity of distinct places leading to incorrect localizations can be observed.

It can be seen that the appearance of distant scenery is nearly unchanged when the camera moves few kilometers back and forth. This means that localization meeting the strict tolerance of 1 kilometer is difficult to achieve in natural environments, leaving room for further research.

References

[1] Touqeer Ahmad, George Bebis, Monica N. Nicolescu, Ara V. Nefian, and Terry Fong. An edge-less approach to horizon line detection. In Tao Li, Lukasz A. Kurgan, Vasile Palade, Randy Goebel, Andreas Holzinger, Karin Verspoor, and M. Arif Wani, editors, *14th IEEE International Conference on Machine Learning and Applications, ICMLA 2015, Miami, FL, USA, December 9-11, 2015*, pages 1095–1102. IEEE, 2015.

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307. Washington, D.C., USA: IEEE Computer Society Press, 2016.

[3] Jan Brejcha and Martin Čadík. GeoPose3K: Mountain landscape dataset for camera pose estimation in outdoor environments. *Image and Vision Computing*, 66:1–14, 2017.

[4] Jan Brejcha, Michal Lukáč, Yannick Hold-Geoffroy, Oliver Wang, and Martin Čadík. LandscapeAR: Large scale outdoor augmented reality by matching photographs with terrain models using learned descriptors. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 295–312, Cham, 2020. Springer International Publishing.

[5] Bingyi Cao, André Araujo, and Jack Sim. Unifying deep local and global features for image search. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 726–743, Cham, 2020. Springer International Publishing.

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[7] Olivier Saurer, Georges Baatz, Kevin Köser, L'ubor Ladický, and Marc Pollefeys. Image Based Geo-localization in the Alps. *International Journal of Computer Vision*, 116(3):213–225, 2016.

[8] Giorgos Tolias, Tomas Jenicek, and Ondřej Chum. Learning and aggregating deep local descriptors for instance-level recognition. In *European Conference on Computer Vision*, 2020.

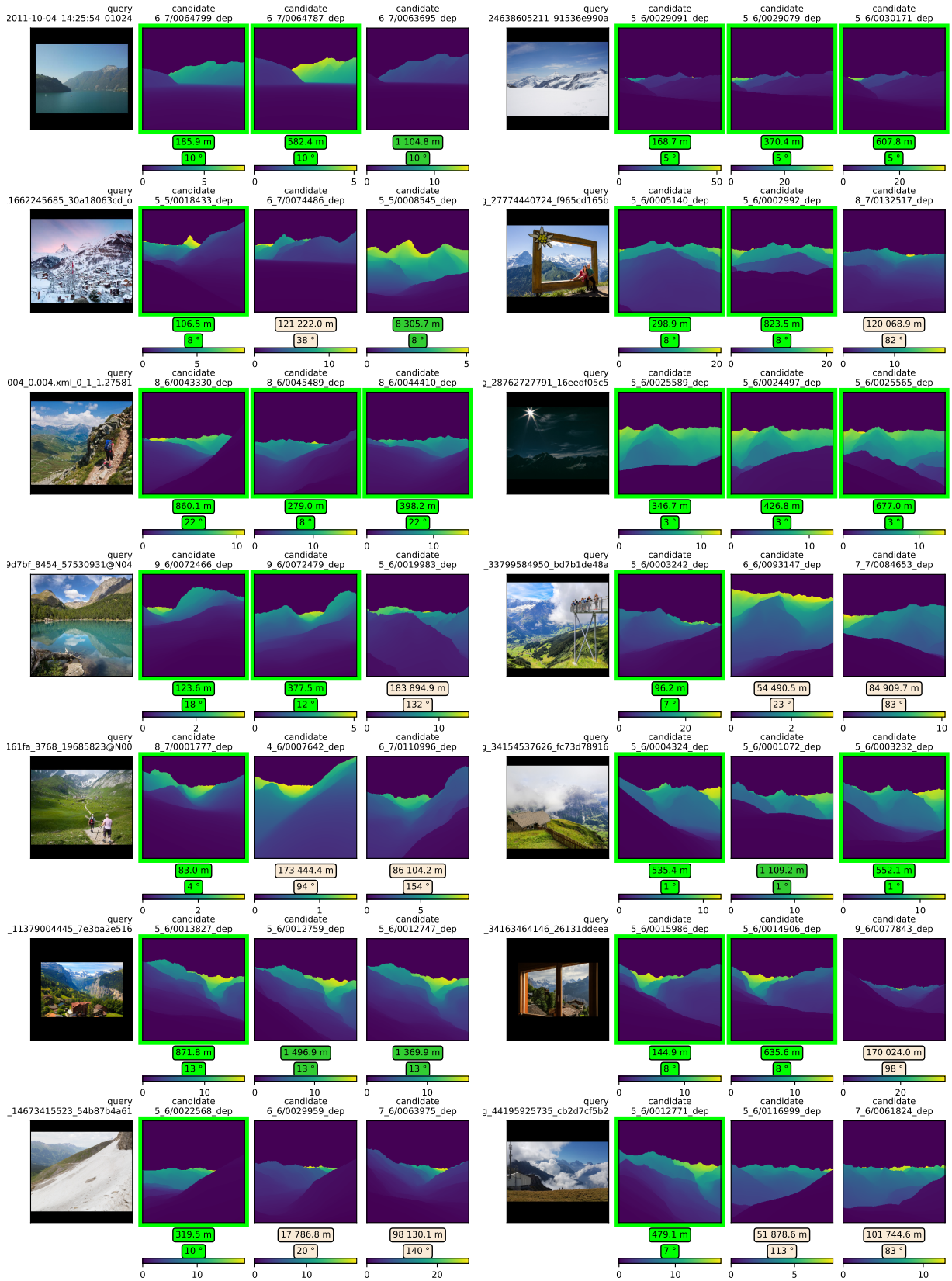

Figure 6. Qualitative evaluation of our CrossLocate approach. A total of 14 **successful** localizations (for 14 query photographs) is shown. Each query is accompanied by its 3 nearest database candidates (depth maps). Candidates within 1 kilometer and 30° in yaw angle are highlighted by a light green border. Candidates within 10 kilometers and 30° in yaw angle are emphasized by a dark green color.

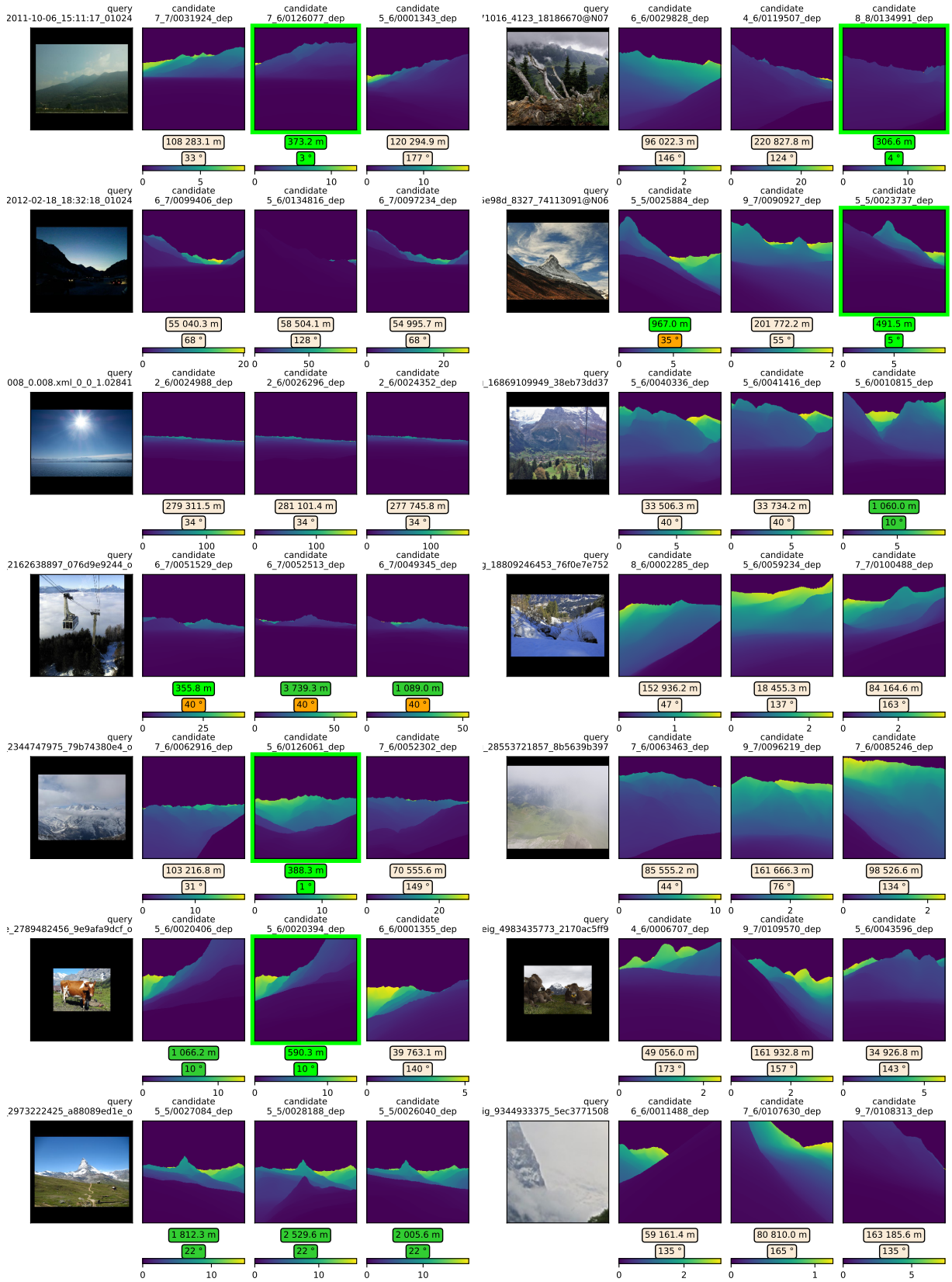

Figure 7. Qualitative evaluation of our CrossLocate approach. A total of 14 **unsuccessful** localizations (for 14 query photographs) is shown. Each query is accompanied by its 3 nearest database candidates (depth maps). Candidates within 1 kilometer and 30° in yaw angle are highlighted by a light green border. Candidates within 10 kilometers and 30° in yaw angle are emphasized by a dark green color.