# Supplementary Material for Semantic Visual Localization

Johannes L. Schönberger[1] Marc Pollefeys[1,3] Andreas Geiger[1,2] Torsten Sattler[1]
[1]Department of Computer Science, ETH Zürich [2]Autonomous Vision Group, MPI Tübingen [3]Microsoft
{jsch,pomarc,sattlert}@inf.ethz.ch andreas.geiger@tue.mpg.de

In this supplementary document, we first give additional implementation details of the proposed semantic visual localization pipeline. Next, we show several examples of semantic query and database maps alongside the obtained localizations and correspondences. Moreover, we provide additional examples of loop closure success and failure cases.

## 1. Implementation Details

**Ours.** We use a batch size of 32 to train our encoder-decoder network for 2,000 epochs using ADADELTA [4] as an adaptive learning rate method for stochastic gradient descent. We set the initial learning rate to $\eta = 1$ without decay and set the hyperparameters to $\rho = 0.95$ and $\epsilon = 10^{-8}$. All convolutional layers use a filter size of $3 \times 3 \times 3$ using zero-padding and *ReLU* activation followed by a $2 \times 2 \times 2$ max-pooling layer. The fully-connected layers are followed by a *tanh* activation function. Upsampling is implemented by repeating the data in the spatial domain by a factor of $2 \times 2 \times 2$. The final convolutional layer of the decoder is followed by a softmax activation. There is a total of around one million learned parameters in our network. For data augmentation, we apply a dropout of 10% on the voxels of the incomplete volume. The reconstruction loss $\Delta_R \in \mathbb{R}$ is emphasized by a factor of 10 relative to the Gaussian prior loss $\Delta_{KL} \in \mathbb{R}$. In addition, for faster convergence during training, the reconstruction loss on occupied voxels is emphasized by a factor of 10. In the experiments, we use 18 uniformly spaced orientation hypotheses $\theta = \{0°, 20°, \ldots, 340°\}$ around the gravity axis.

**SIFT.** For SIFT feature detection, we use 4 octaves starting with a two times up-sampled version of the original image, 3 scales per octave, a peak threshold of $\frac{0.02}{3}$, an edge threshold of 10, and, due to the gravity-aligned input, an upright orientation assumption. The visual vocabulary is represented by $2^{16}$ visual words embedded in a $N_B = 64$ dimensional Hamming space and using a hierarchical branching factor of 256. Using these settings, we obtain several thousand descriptors per image. Localization is performed using a traditional image retrieval setup with two-view geometric verification on the top-ranked retrievals and 2D-3D camera pose estimation inside RANSAC followed by a non-linear refinement. A camera pose is considered as verified, if it has at least 15 2D-3D inlier correspondences. The 3D map is obtained through fusion of all database depth maps. Similar setups achieve state-of-the-art results [2, 3].

**DSP-SIFT.** We use the same feature detector as for standard SIFT and a total of 10 pooling scales uniformly spaced between $\frac{1}{6}$ and 3. We train a new visual vocabulary for nearest neighbor search. Otherwise, we use the same setup as for standard SIFT.

**MSER.** Using the same setup as for DSP-SIFT, we replaced the SIFT keypoint detector with MSER using a step size between 2 intensity threshold levels, region sizes between 30 and 14000 pixels varying by a maximum of 25%. We extract DSP-SIFT descriptors as described previously for the detected regions and train a new visual vocabulary for nearest neighbor search.

**VIP.** For our VIP experiments, we manually selected road regions in multiple images from different viewpoints, fitted a plane through the 3D road points, normalized image to a fronto-parallel viewpoint, densely extracted SIFT descriptors, and matched them exhaustively between the images. For all but very similar viewpoints, we failed to establish correct correspondences between the images. Our main insight from this experiment was that the geometric and radiometric distortions are too severe for low-level appearance matching. We therefore excluded VIP from the further evaluation.

**DenseVLAD.** Using the same setup as for DSP-SIFT, we extract a 4096 dimensional global image descriptor using DenseVLAD, which replaces the visual vocabulary based image retrieval pipeline. To find nearest neighbor images for a given query images, we exhaustively compare the global image descriptors from the query to the database and then perform two-view geometric verification on the top-ranked retrievals, equivalent to the DSP-SIFT experiment.

**FPFH.** Using the same keypoint locations and geometric verification approach as for our method, we extract 33 dimensional FPFH descriptors and train a new vocabulary for nearest neighbor search.

**CGF.** Equivalent to FPFH, we replaced our learned descriptors with 32 dimensional CGF descriptors and train a new vocabulary for nearest neighbor search. We consistently oriented the point cloud normals between the query and database maps towards the cameras.

**3DMatch.** For this experiment, we tried both the pre-trained 3DMatch models and also fine-tuned the descriptor using corresponding complete and incomplete subvolumes that we also used to train our descriptor. The fine-tuned model performs slightly better and we use it for our experiments. Equivalent to FPFH, we then replaced our learned descriptors with 512 dimensional 3DMatch descriptors and train a new vocabulary for nearest neighbor search.

**PoseNet.** For each database, we train a separate PoseNet model from scratch until convergence, which required more than 2 days of training for the largest models. We then regress the pose for each query image, which serves as the single, top-ranked pose hypothesis for the evaluation.

**DSAC.** For this experiment, we trained DSAC from scratch using the suggested initialization protocol, which took around 2 days for the smallest KITTI odometry sequence 04. However, we could not produce meaningful pose estimates and our main insight from investigating the issue was that the current DSAC approach has problems with repetitive structures and larger scale outdoor scenes. We therefore excluded DSAC from the further evaluation.

## 2. Localization Results

**KITTI** Figs. 1, 2, and 3 visualize localization results for three KITTI odometry sequences. The semantic maps have been built by fusing all images, depth maps, and semantic segmentations of the left camera in a sequence. The depth maps were computed by two-view stereo between the left and right camera using semi-global matching [1]. The images, depth maps, and semantic segmentations are jointly fused into semantic 3D maps, which are stored in an efficient Octree data structure at a maximum leaf node resolution of 0.3m. Visualized are all leaf nodes and their corresponding most likely semantic class labels. In addition, we show one example of the different loop closure scenarios and two hard cases caused by ambiguities. For the $90^\circ$ scenario, we excluded all images from the database that are not within a viewpoint change of $90^\circ \pm 20^\circ$ w.r.t. the query image. Equivalently, for the $180^\circ$ scenario, we excluded all images from the database that are not within a viewpoint change of $180^\circ \pm 20^\circ$ w.r.t. the query image. Local ambiguities arise when there are multiple repeating structures in close vicinity. Global ambiguities are rarer and are caused when different parts of the map looks similar both in terms of geometry and semantics. Note that despite ambiguous correspondences between query and database map, our proposed alignment and verification procedure is almost always able to determine the correct location of the query. Fig. 4 shows several alignments between the query and database map, which were obtained using our localization pipeline. Note that the query is aligned accurately even in the case of missing observations and noise.

**NCLT** Figs. 6 and 7 show localization results for the NCLT dataset. Opposed to KITTI we use the LIDAR point cloud and camera 5 of the Ladybug rig for 3D semantic fusion. We use the same descriptor trained on a KITTI-like dataset and, otherwise, use the same setup as for KITTI, demonstrating that our method generalizes across different scene types and different sensors without re-training. Note that NCLT contains both extreme seasonal/illumination changes as well as extreme viewpoint changes between the different datasets. Our method is able to localize robustly in this challenging scenario.

## 3. Local Feature Correspondences

Fig. 5 visualizes corresponding volumes between the query and database maps. The correspondences were established through nearest neighbor search using our proposed semantic descriptor and vocabulary. Note that our descriptor is robust

to missing observations and significant noise arising from inaccuracies in depth estimation, semantic segmentation, and 3D fusion, which form the input to our method.

## References


[1] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 30(2):328–341, 2008. 2

[2] T. Sattler, M. Havlena, F. Radenovic, K. Schindler, and M. Pollefeys. Hyperpoints and fine vocabularies for large-scale location recognition. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2015. 1

[3] T. Sattler, M. Havlena, K. Schindler, and M. Pollefeys. Large-Scale Location Recognition And The Geometric Burstiness Problem. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[4] M. D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv.org*, 2012. 1

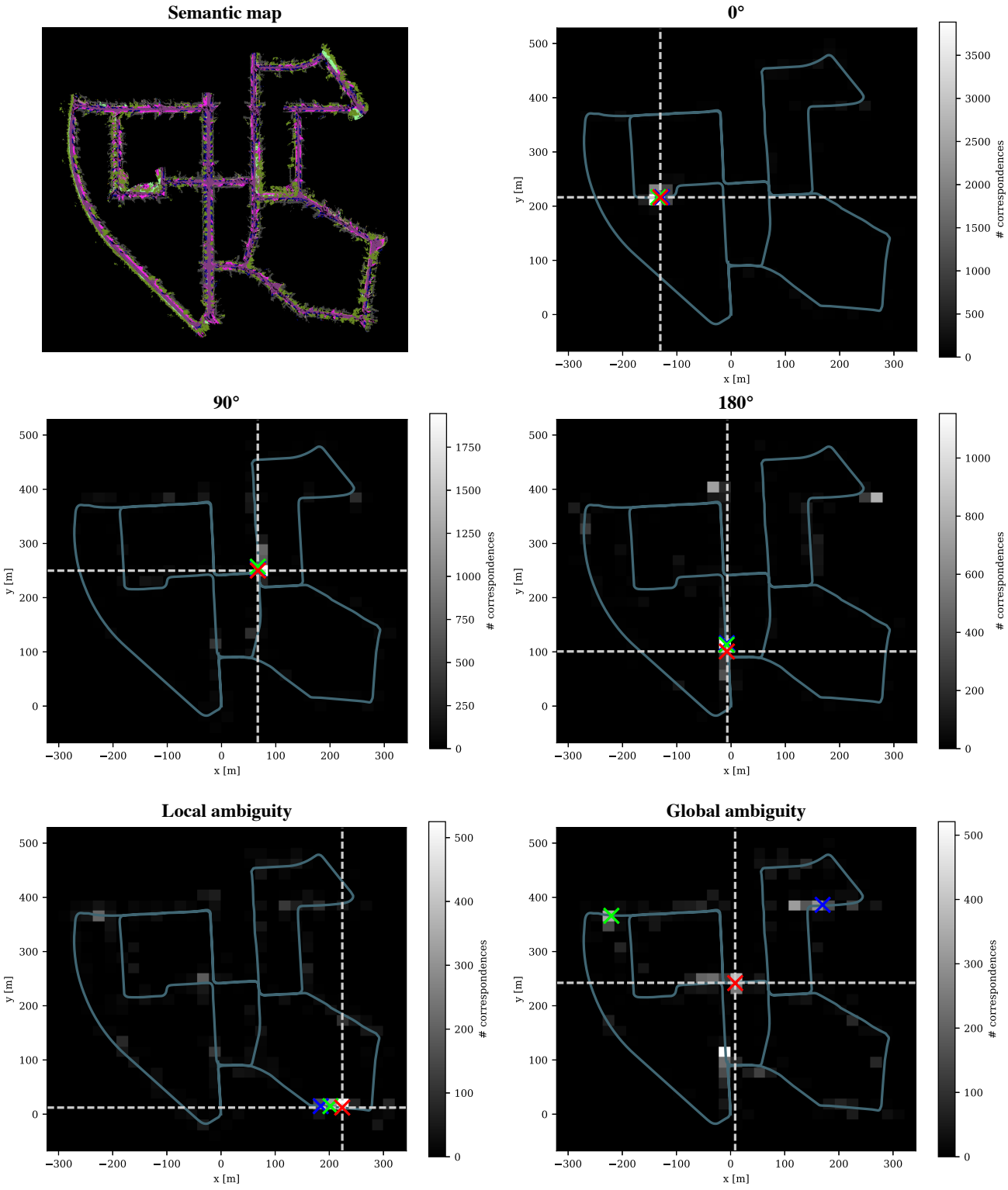


Figure 1: Localization results for sequence `00` of the KITTI odometry dataset. Top-ranked localization results visualized with crosses (red: 1st, green: 2nd, blue: 3rd). Ground-truth location visualized with dotted line. The background histogram visualizes the distribution of corresponding volumes in the database map. Note that, even in the presence of ambiguities, our spatial verification is able to localize correctly.

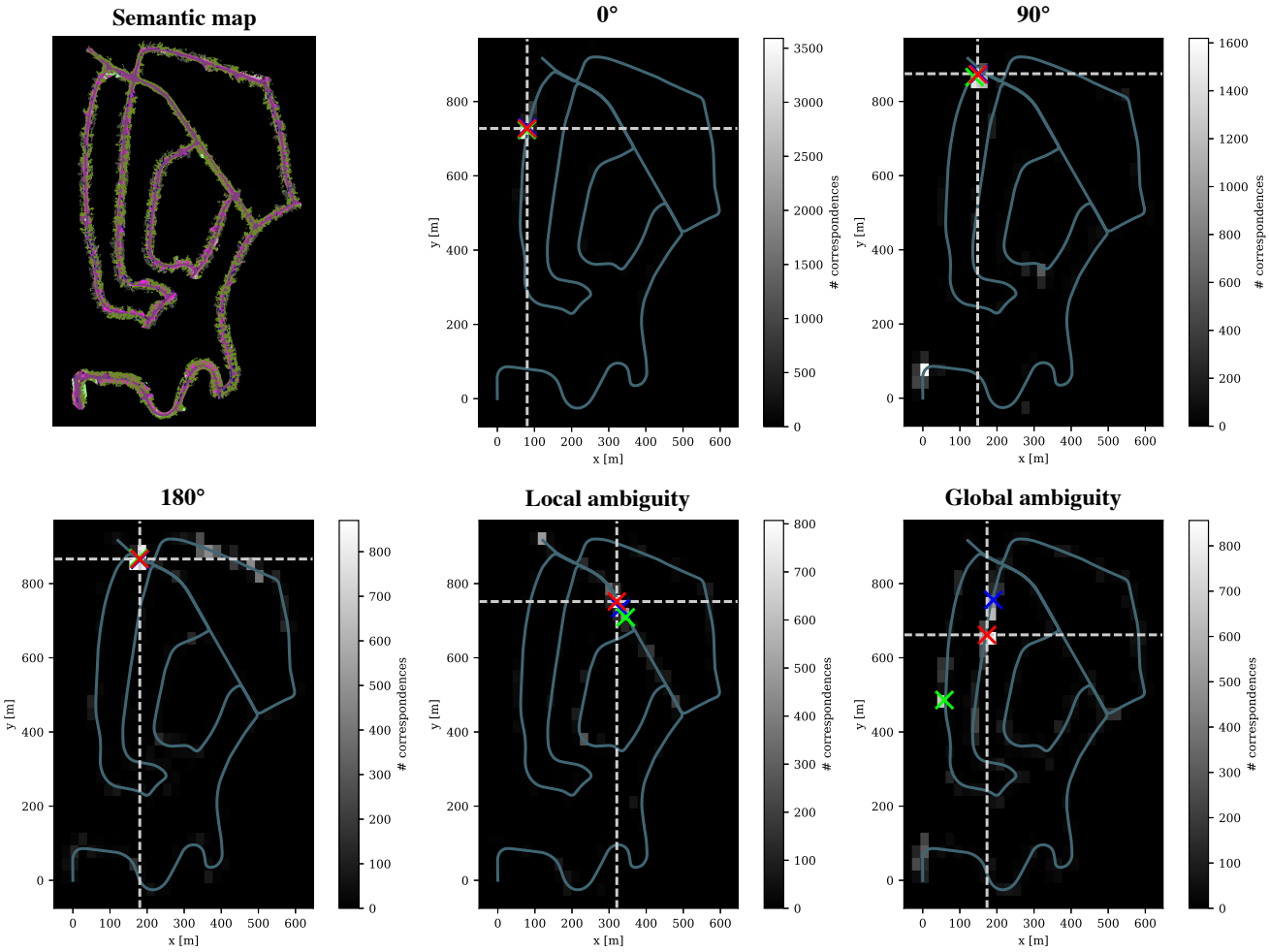


Figure 2: Localization results for sequence `02` of the KITTI odometry dataset. Top-ranked localization results visualized with crosses (red: 1st, green: 2nd, blue: 3rd). Ground-truth location visualized with dotted line. The background histogram visualizes the distribution of corresponding volumes in the database map. Note that, even in the presence of ambiguities, our spatial verification is able to localize correctly.

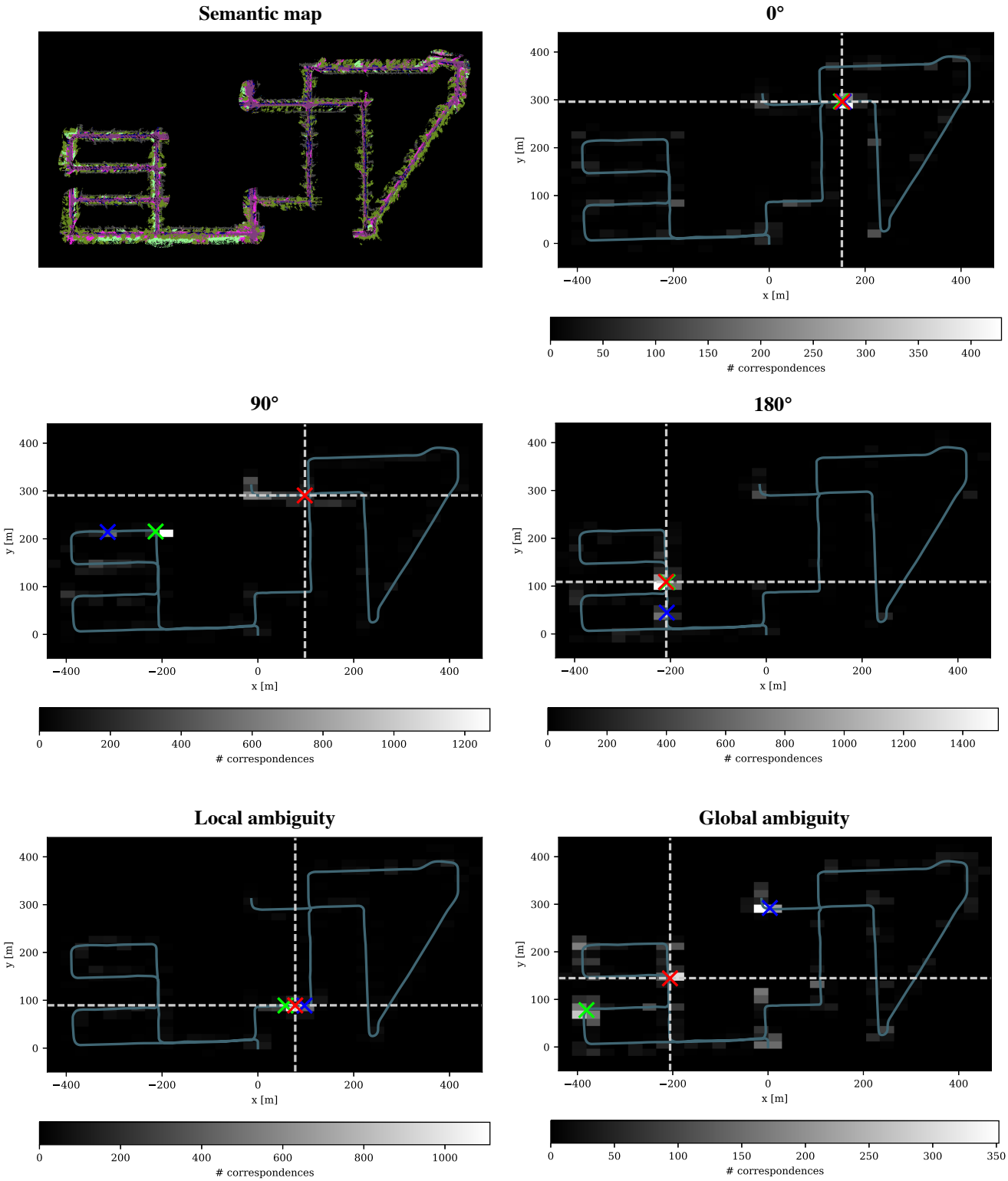


Figure 3: Localization results for sequence `08` of the KITTI odometry dataset. Top-ranked localization results visualized with crosses (red: 1st, green: 2nd, blue: 3rd). Ground-truth location visualized with dotted line. The background histogram visualizes the distribution of corresponding volumes in the database map. Note that, even in the presence of ambiguities, our spatial verification is able to localize correctly.

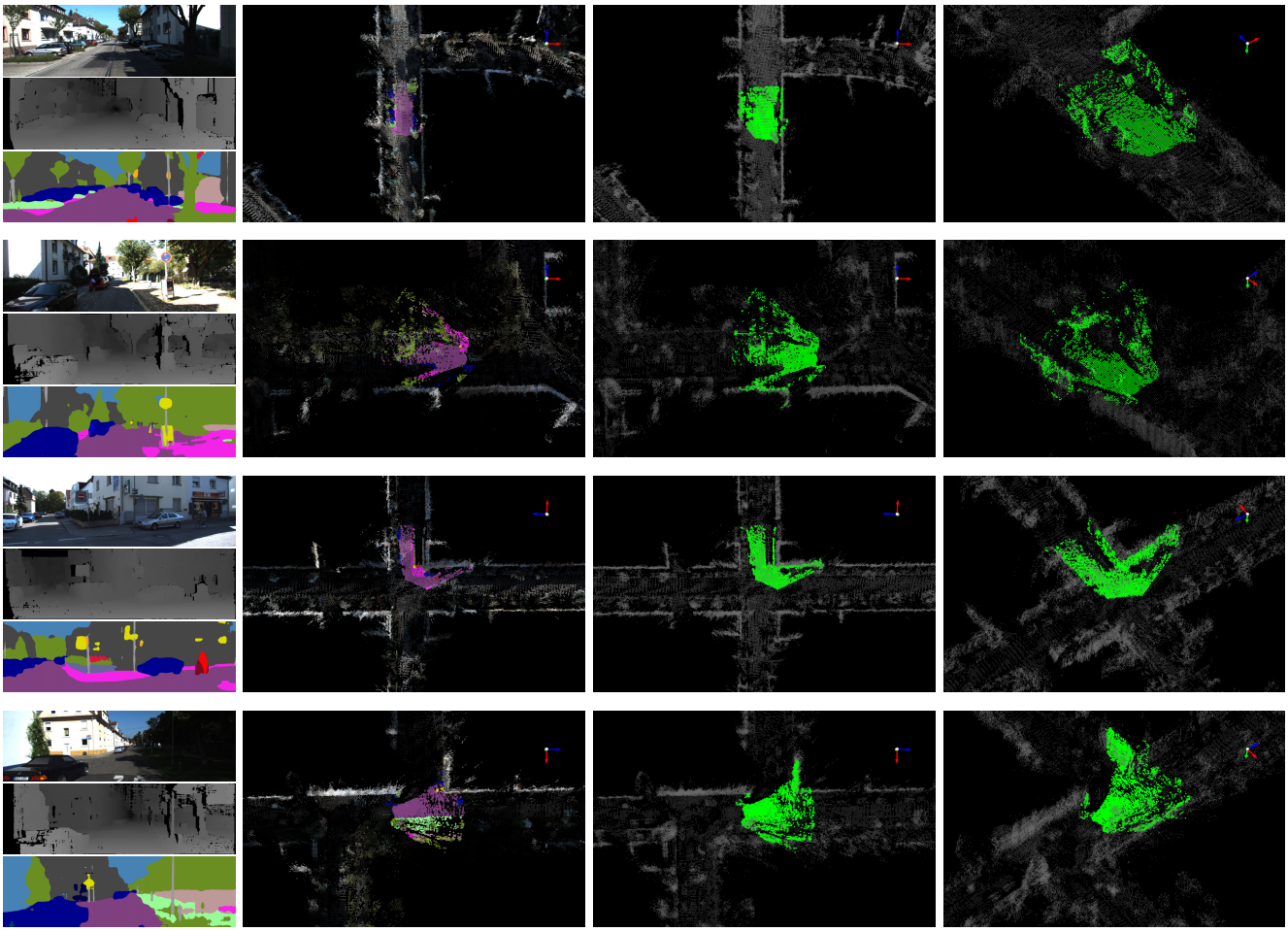

Figure 4: The left column shows the input image and its corresponding depth map and semantic segmentation. The second column shows a top-down view of the aligned query and database map using semantic coloring for the query and RGB coloring for the database. The third column shows the same view but using green color for the query and gray for the database map, while the last column shows the same but from a different viewpoint. Note that the query and database maps are aligned accurately even in the presence of missing observations and significant noise.

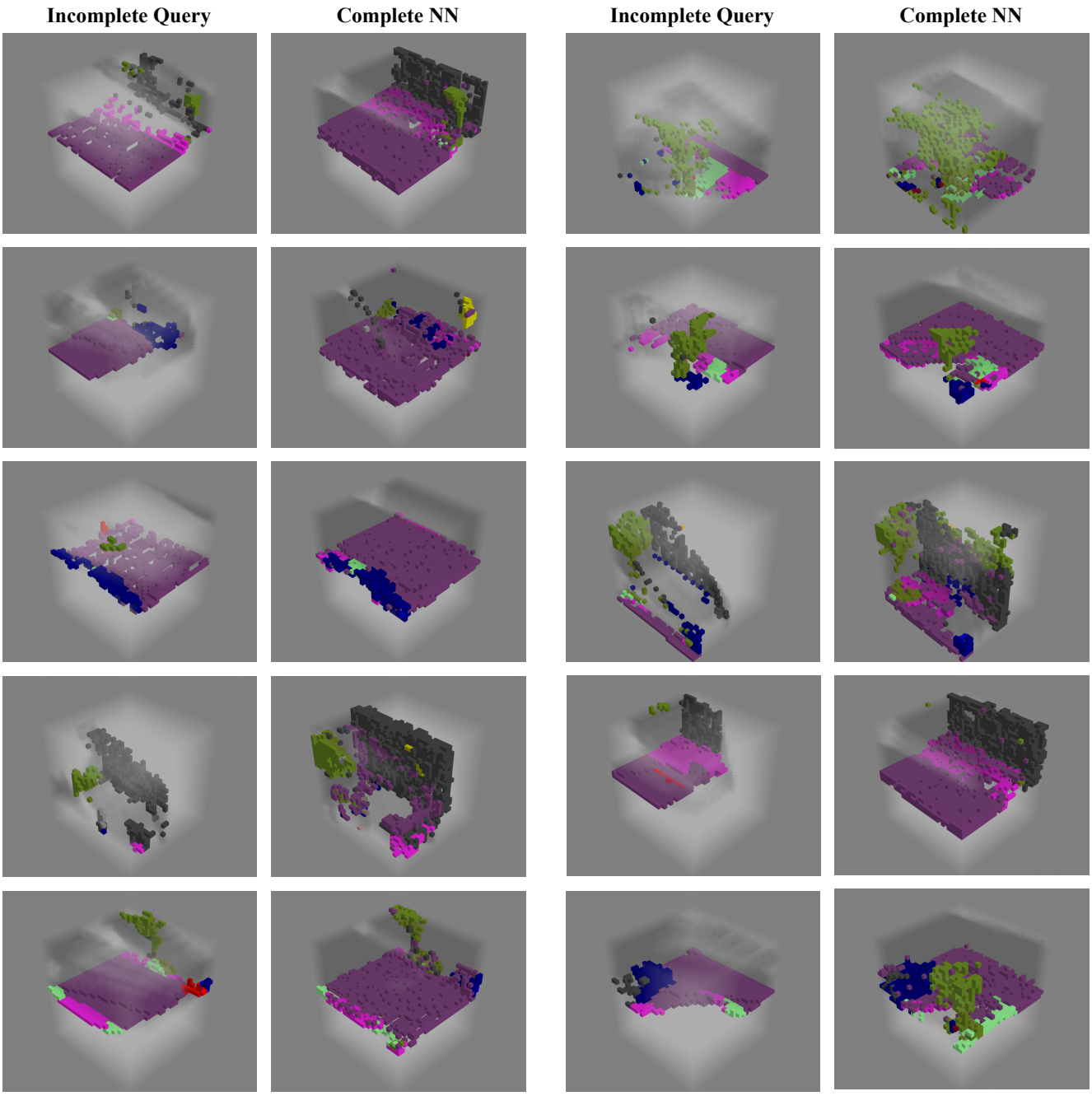


Figure 5: The corresponding volumes in the query map (incomplete query) and retrieved nearest neighbor volumes (complete NN) in the database map. Nearest neighbor search was performed with our proposed semantic vocabulary tree.

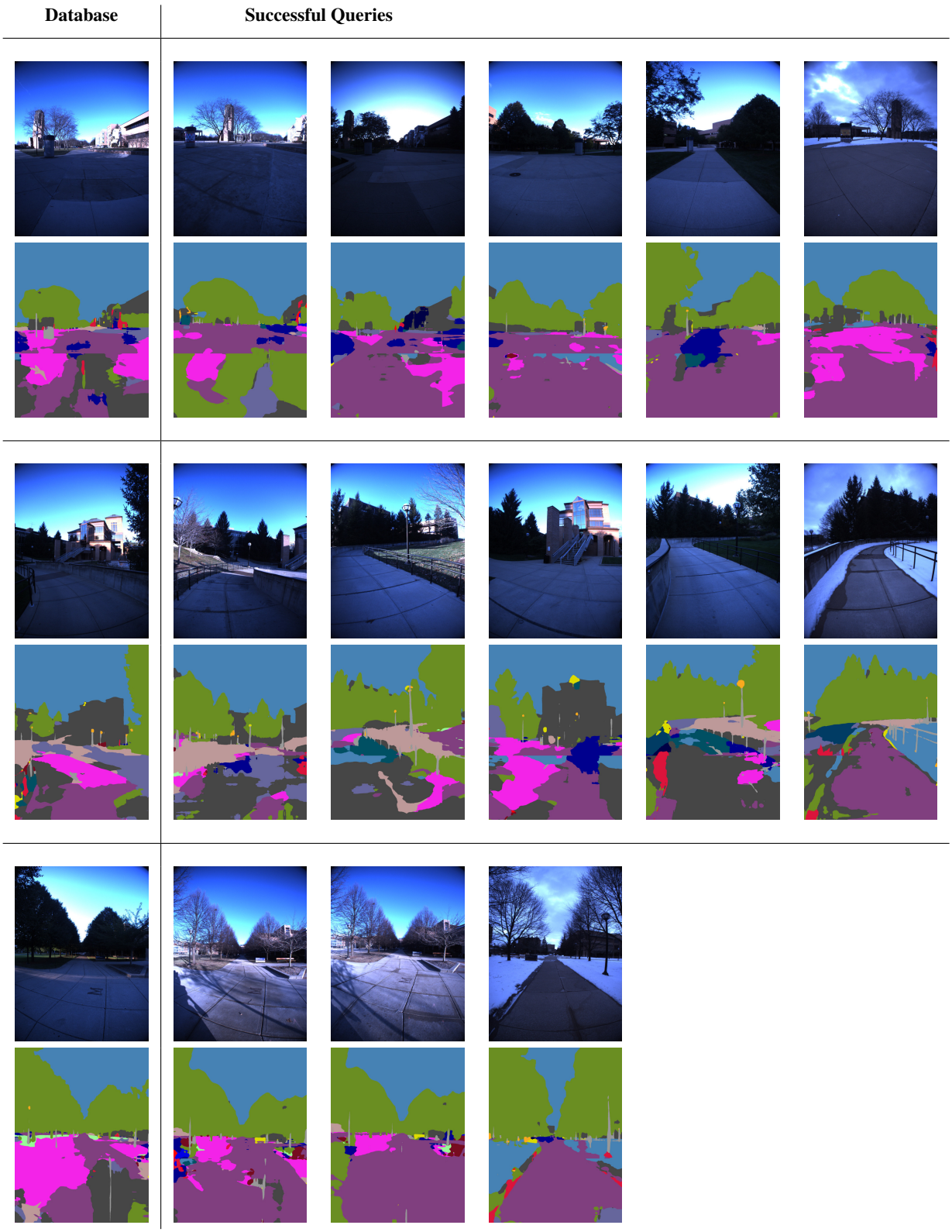


Figure 6: Localization results for NCLT dataset. Left-most image depicts database scene, while images to the right show successful localization results under different viewpoints and illumination/season. Top row shows RGB images while bottom row shows corresponding semantic segmentation. Results continued on next page in Figure 7.

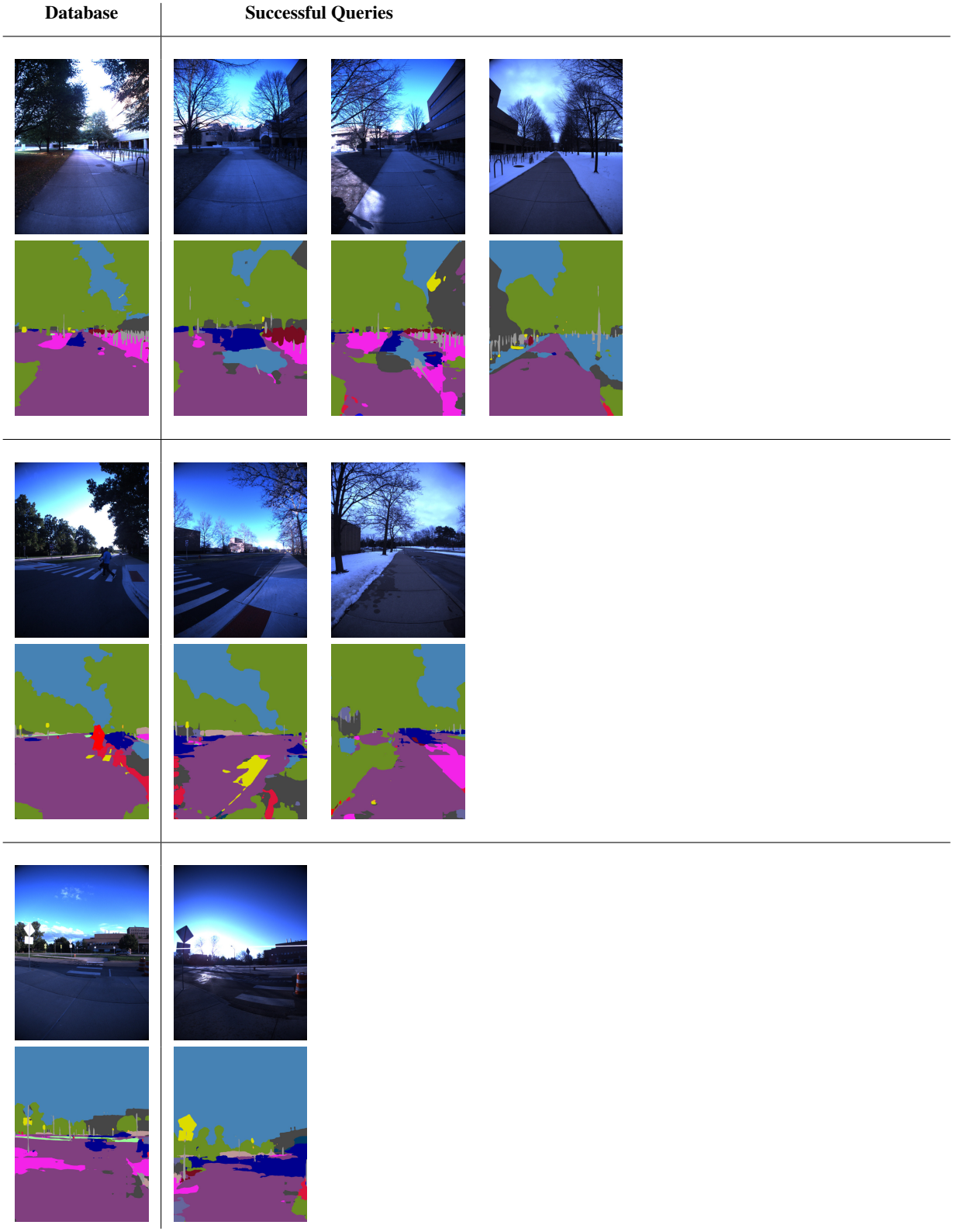


Figure 7: More localization results for NCLT dataset, continued from Figure 6.