SegLoc: Learning Segmentation-based Representations for Privacy-Preserving Visual Localization

Maxime Pietrantoni[1,2] Martin Humenberger[3] Torsten Sattler[2]
Gabriela Csurka[3]
[1] Faculty of Electrical Engineering, Czech Technical University in Prague
[2] Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague
[3] NAVER LABS Europe
{firstname.lastname}@cvut.cz,{firstname.lastname}@naverlabs.com

Supplementary

In this supplementary material, we first provide details about the training data preparation (Sec. 1.) as well as the training process (Sec. 2.) In Sec. 3. we present experiments exploring the accuracy-memory trade-off while we add further ablative studies in Sec. 4. (and interpretation of the ablative experiments of Sec. 5.1. in the main paper). In Sec. 5 we investigate how nonlinear downprojection of features would compare to our method. Sec. 6 provide more insight regarding the convergence of the pose refinement. Finally further visualizations may be found in Sec. 7 .

1. Training data preparation

Training SegLoc requires image-level information in the form of set of anchor/positive/negatives images as well as pixel-level information in the form of dense correspondences. We train our models both on outdoor (ECMU Seasons [1, 16]) and indoor environments (Indoor6 [5]). We obtain the training data automatically using only the images and GT poses provided in the official release of these dataset without any additional ground truth information or manual processing. Our approach to creating the training data is described below and follows the approach proposed by [9].

ECMU. The Cross-Seasons Correspondence Dataset (CSC) [9] was built upon the training slices (7-12 and 22-25) of the Extended CMU (CMU) seasons [1, 16] dataset and contains image pairs with 2D-2D correspondences. Aiming to extend this dataset with more weather diversity and stronger viewpoint changes, we proceed as follows on each training slice independently. We use the kapture pipeline [6] to compute R2D2 [7] local descriptors in the images. Then merging the image sets from various weather conditions, we build a sparse SfM model with COLMAP [14] by triangulating the 2D matches using the camera poses followed by building a dense model using COLMAP's multi-view stereo pipeline [15]. The resulting dense point cloud is split into sub-point clouds, each of them being associated to a specific weather condition (based on the condition labels of the training images provided by the dataset). A 3D point is associated to a sub-point cloud if it is observed by at least three images captured under that given weather condition. Then, given a pair of sub-point clouds, 3D-3D correspondences are established by finding mutual nearest neighbors. Reprojecting these points into the images yields a list of 2D-2D correspondences for all image pairs that are part of the sub-point clouds. Note that this 3D geometric matching step does not provide any guarantee with regard to the validity of the correspondences. Therefore, we reject all 3D-3D correspondences whose reprojection error is above 5 pixels. Finally, image pairs with less than 500 correspondences are eliminated.

Indoor6. For each Indoor6 scene, we apply the same pipeline to compute images pairs and correspondences. The only differences being that we build a sparse SfM model from SIFT keypoints and we do not explicitly split the dense point cloud depending on capture condition as the scene is not evenly covered by each capture condition. Thus for an image, the candidate image pair is searched among the whole image set. Our global representations are spatially pooled from the dense segmentation which is equivariant wrt. viewpoint change. As the global representations must show some level of invariance to viewpoint change it is preferable that training image pairs have limited viewpoint change and sufficient visual overlap Furthermore, enforcing dense consistency only within a small region of the segmentation would not providing sufficient learning signal.

Hence, the bounding box containing all 2D points within the first image is reprojected in the second image and vice versa. Overlap ratios between the reprojected bounding boxes and images are computed and used to select pairs with a sufficient correspondence coverage eliminating pairs

below a threshold of 0.75. No constraint regarding the relative pose between images is enforced on ECMU. However, on Indoor6 we discard pairs with relative rotation difference superior to 25 degrees. In Table A.1 we summarize the number of retained pairs per cross-weather condition on ECMU (summed over all training slices) or per scene on Indoor6. The aim of including such a weather/capture conditions diversity among the training data was to increase the robustness of the learned representations. In addition, in Figure A.1 we display example image pairs along with their retained correspondences and the containing bounding boxes showing the shared regions.

	Weather 1	Weather 2	Number of pairs
ECMU Seasons	Overcast Foliage	Cloudy Mixed Foliage	2167
	Sunny Foliage	Cloudy Foliage	1141
	Overcast Mixed Foliage	Overcast No Foliage	5040
	Cloudy Low Sun	No Foliage Snow	8120
	Overcast Foliage	Low Sun Foliage	1970
	Sunny Foliage	Low Sun Foliage	7400
	Low Sun Mixed Foliage	Sunny No Foliage	915
	Overcast Foliage	Sunny No Foliage	5520
Indoor6	Scene 1		4349
	Scene 2a		7811
	Scene 3		2166
	Scene 4a		1491
	Scene 5		3333
	Scene 6		848

Table A.1. Training image pair distribution for the ECMU Seasons and Indoor6 datasets.

2. Implementation details

Unsupervised initial clustering. The initial clustering is a key element of the model because, as described in Sec. 3. in the main paper, the derived prototypes play several roles in our framework. Indeed, in the first epoch they are used to derive the pseudo targets to train the classifiers, thus ensuring a good initialization of the discriminative clustering phase (see Sec. 3.1. of the main paper). Furthermore, the prototypes help regularizing the training process by incorporating some initial prior semantic structure in the feature space (see Sec. 3.2.). Therefore, to ensure a good initialization, similar to [10], we consider an available pre-trained semantic segmentation model and use it to extract and cluster per-pixel features considering a random subset of the training set (reference images). Using a pre-trained segmentation model guarantees some meaningful features for the clustering.

Concretely, we use the DPT-hybrid [12] model pre-trained on the ADE20k segmentation dataset [19] as our initial model. ADE20k is a semantic segmentation dataset containing 150 classes covering both outdoor and indoor scenes. In the initialization step, reference images are processed by the initial SegLoc network (where the parameters are initialized with the pre-trained DPT) and dense features from the encoder are sampled and their associated ADE20k predictions are collected. We then group the dense features according to their ADE20k predictions (removing ADE20k classes with low population, corresponding in general to indoor semantics when we process ECMU and outdoor semantics when we process Indoor6). Within each remaining ADE20k class we apply sub-clustering using either K-means or Meanshift [4]. Our final set of initial prototypes $\{\mathbf{c}\}_{k=1}^{K}$, is the union of these sub-cluster centers. Note that we select the parameter k of K-means respectively the Meanshift's bandwidth such that the total number of prototypes equals the desired granularity K of the segmentation. This initial clustering step is applied independently on each level l of the hierarchical decoder yielding four sets of initial prototypes, which are then refined during training to leverage information from different spatial granularity.

Algorithm 1 The full model training.

Data: Image pairs I^a, I^b with 2D-2D correspondences $\{x^a_{u_l}, x^b_{v_l}\}$
Initialize the encoder with DPT [12] pre-trained on ADE20k
Generate the K prototypes $\mathbf{c}_k$
for *epoch in range N_epoch* **do**
  Sample features $\mathbf{F}_i^j$ from reference images I_j and compute class predictions $\mathbf{p}_i^j$
  Compute empirical distribution $\mathbf{d}^p$ using $\sigma(\mathbf{p}_i^j)$
  Update the prototypes $\mathbf{c}^k$ with the features $\mathbf{F}_i^j$ and compute the cluster concentrations ϕ^k
  for *each batch in epoch* **do**
    Network forward pass
    if *epoch==1* **then**
      update q_{in} with (4)
    else
      update q_{in} with (2)
    end
    Update parameter network θ, ϕ minimizing the overall loss
    $\mathcal{L} = \mathcal{L}_{CE} + \mathcal{L}_{CC} + \mathcal{L}_{PC} + \mathcal{L}_{PF} + \mathcal{L}_{MS}$
  end
end

Training details. As the segmentation backbone for our model we rely on DPT [12], which has a hierarchical encoder/decoder architecture with vision transformers and convolutions (see Figure A.2). After the initial clustering step, we replace the classification head of each decoder level of the pre-trained model with a randomly initialized MLP, followed by batch normalization. The number of target classes is set to K. By default we use $K = 100$ (but as shown below in Sec. 4., we also evaluate and compare our model using coarser or finer segmentations by varying K). The dimension of the feature map $\mathbf{F}$ is set to $D = 256$. We use the Adam optimizer [8] with an initial learning rate of 2e-3 and a 1e-4 weight decay. For outdoor environments we

Figure A.1. Example pairs of images, along with 2D-2D correspondences between them, from the dataset used to train our approach.

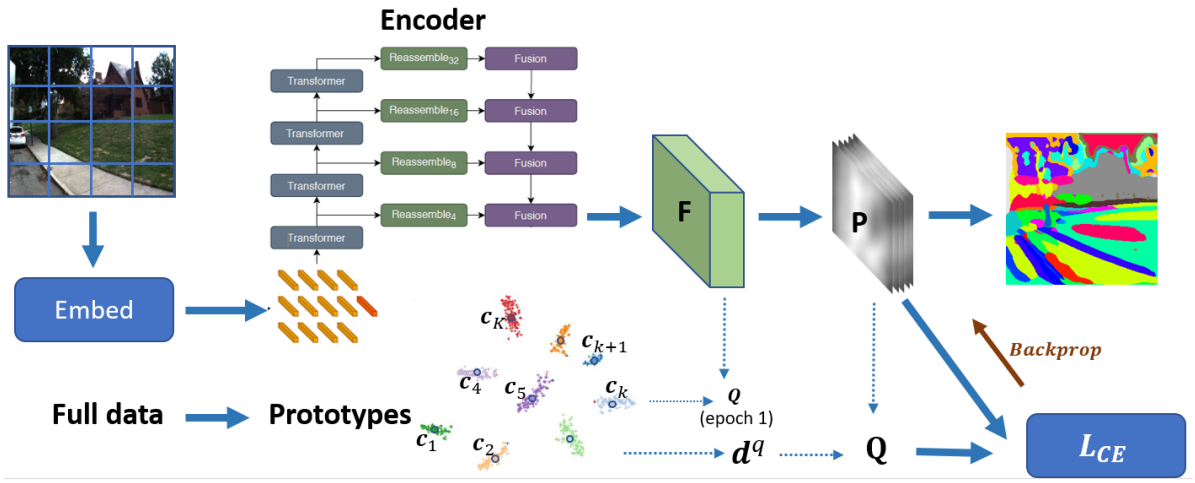

Figure A.2. Our self-supervised segmentation network architecture. Prototypes/feature similarities are used as targets during the first epoch. In the following epochs, the target distributions act as pseudo-labels to guide the segmentation.

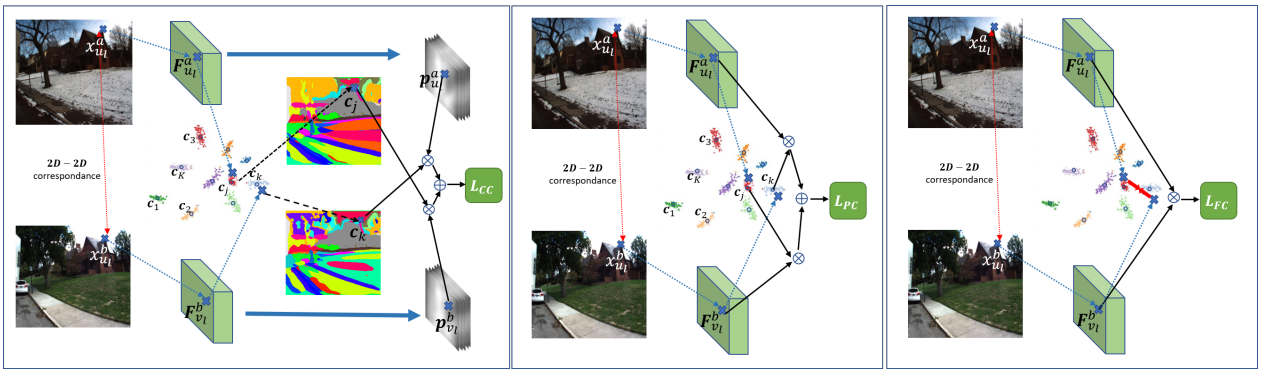

Figure A.3. We illustrate here our consistency losses. Given a pixel-to-pixel correspondence and a set of prototypes, we enforce consistency between the representations while infusing prior information from the prototypes. Column 1 : Correspondence consistency loss Column 2 : Prototypical cross contrastive loss, Column 3 : Contrastive feature consistency loss.

finetune our model on the pose refinement task of ECMU training set. Training is distributed among 2 Nvidia V100 GPUs with a total batch size of 4, each GPU processing 2 image pairs along with 2 positive and 4 negative associated images selected as follows per iteration (needed for the MS Loss $\mathcal{L}_{MS}$). Given an anchor image, we mine one hard positive, one negative, and one hard negative. The hard positive is the least similar image within a given radius (5 meters for ECMU and 1.5 meters for Indoor6) from the anchor's position. The hard negative is the most similar image not located within a given distance (10 meters for ECMU and 3 meters for Indoor6) from the anchor's position. By similarity between images we refer to their global descriptors' cosine similarity. These descriptors are computed and stored at the beginning of each epoch. During training, images are resized to 720 $\times$ 720 pixels and randomly cropped and rescaled to a 640 $\times$ 640 pixels size. The associated correspondences' coordinates are adapted to compensate for rescaling and crop caused by these image transformations. Random photometric transformations including color jittering, Gaussian blur, and gray scaling are also applied during training as data augmentation. During inference, images are resized to 960 $\times$ 960 pixels size.

In Algorithm 1, we summarize the steps of the full training procedure of our model. Fig A.2 illustrates the discriminative clustering process which is casted as classification task. Pseudo targets Q are derived and used in $\mathcal{L}_{CE}$ to learn the classifiers (Sec. 3.1. of the main paper). Additionally, Fig. A.3 illustrates the behavior of the consistency losses ($\mathcal{L}_{CC}, \mathcal{L}_{PC}, \mathcal{L}_{PF}$) described in Sec. 3.2. of the main paper.

Details about evaluating privacy.

To evaluate qualitatively and quantitatively how privacy preserving different models are (*c.f.* Sec. 5.2 of the main paper), we use the SfM inversion pipeline from [11] to recon-

struct images from the map. The pipeline contains three networks. Given a database 6dof pose and a database 3D sparse model, the pipeline tries to reconstruct the associated database image. The first network predicts the visibility of the 3D points from the perspective of the image's camera. We are interested in showing whether it is possible to recover an image from underlying 3D representations. These representations are either a single SegLoc segmentation label or Pixloc visual descriptor associated to each 3D point of the 3D model. For this task, we are interested in the worst-case scenario, i.e., the attacker has access to visibility information. Hence, we did not use nor implement this visibility network. We re-implemented the rest of the pipeline. We train the CoarseNet and RefineNet inversion models which consist of U-Nets with encoder / decoder layers and symmetric skip connections. CoarseNet takes as input a sparse tensor, representing the image to be reconstructed, where the visible 3D underlying representations are inserted into this tensor at their reprojected locations. The model then tries to reconstruct the associated RGB image. RefineNet takes as input the concatenation of the output of the first U-Net with the sparse input tensor and also tries to reconstruct the image. These networks are trained with L1 losses, perceptual losses [18], and a discriminator with a BCE loss. They are trained for 20 epochs with the Adam optimizer and an initial learning rate of 1e-4. For both Pixloc and SegLoc, we use only the finest representation maps with the highest resolution as descriptors for the inversion experiments.

3. Storage requirement *vs*. accuracy trade-off

We have shown in the main paper that using a single cluster label per keypoint (database side) and pixel (query side) allows SegLoc to be privacy preserving and storage efficient. In this section, we explore pose refinement when instead of single cluster label, we store and use part of or the full distribution over the cluster set.

To better quantify the Storage requirement *vs*. accuracy trade-off, we run the refinement procedure with the following choices for point representations. Both on the database and query side, given a class probability vector $\mathbf{p}_i$, we keep the top-K highest logits for $K = 1, 3, 5, 10, 25, 50, 100$ (here, K=100 corresponds to using the full probability vector). We run the refinement experiments on the ECMU test slices 6, 13 and 21. When experimenting with the database points representations, we use the full class probability vector for query points. When experimenting on the query representation we keep using a single one-hot encoded label for database points. Input images are resized to 480 × 480 and perform refinement using a single segmentation head and no coarse-to-fine refinement. We report the percentage of queries localized within (.25m/2°) and (.5m/5°) for each input representation in Fig. A.4.

As expected, increasing the amount of input information increases the localization accuracy. However it comes at the cost of increased memory footprint. Adding more than 10 labels yields relatively lower accuracy improvement compared to the first labels. It confirms that our dense representations are discriminative and only a small part of the class probability vector is actually required to run the refinement effectively. Selecting only the top-k predicted class likelihood leads to better storage requirement-accuracy trade-off and the labels can be stored with low storage footprint. It underlines our choice of leveraging a single cluster label during the pose refinement. While the drops between top-1 and top-3 in Fig. A.4 is relatively important, we expect much smaller drop in the case of the full resolution hierarchical model (the difference between single label (topk=1) and full representation (topk=100) is already much smaller in SegLoc vs SegLoc full in Tabs. A.2 and A.3 (database side) or SegLoc vs SegLoc SL in Tab. 2. and Tab. 3. of the main paper (query side)).

Next, we consider refinement where instead of storing a single label per 3D point we store the full distribution (top-k=100). Similarly we use the full distribution in the query (top-k=100). The experimental configuration is the same as in the main paper (960*960 image input resolution, hierarchical model). We report results for ECMU in Tab. A.2 and for Indoor6 in Tab. A.3. We also added comparative results from the main paper.

As expected, increasing the richness of the input representations induces a clear improvement in localization accuracy which in turn comes at a cost of reduced privacy and much higher storage requirement. In this setup, SegLoc bridges the performance gap with keypoint based method.

4. Ablations

Here we complement our discussion about ablating the different components of our model and provide additional ablation studies. These experiments were realized at an initial stage of the project where we used 480 × 480 resized images and only a single classification head (instead of hierarchical decoder) both for training and inference. As such, they were used to validate the architectural choices of the method section and to guide the design/training of our final models. Note however, that we expect the conclusions drawn from these experiments to remain valid for the final models as the core components do not change, as suggested by our results.

Ablating the role of different losses. In Tab. 6, we study more in depth the impact of the individual components of our approach on its performance (*c.f.* Sec. 5.1. of the main paper). As can be seen, all losses contribute to the overall performance and improve upon the core model using only the discriminative clustering (first row). The feature consistency loss $\mathcal{L}_{FC}$ has the most impact as it explicitly

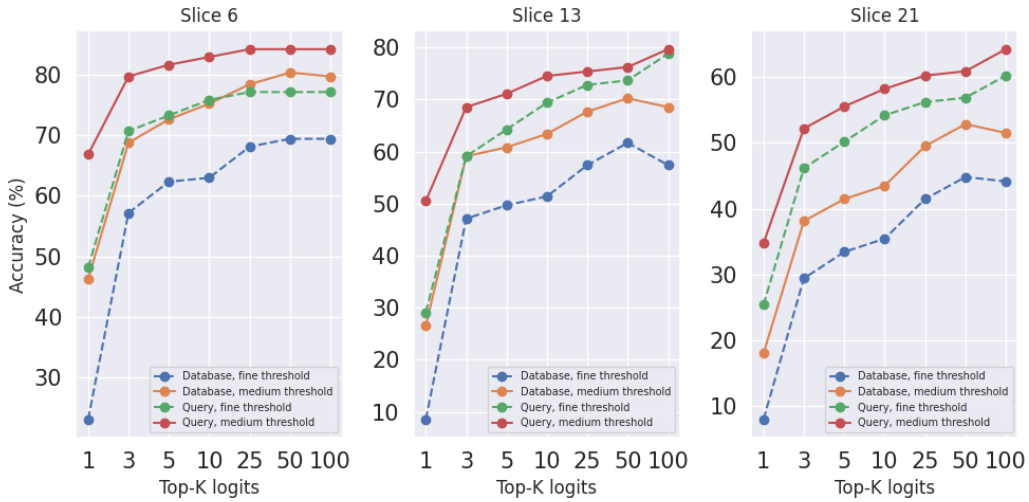

Figure A.4. Fine (dotted line) and medium (straight line) accuracy after refinement for different input representation (database and query side).

		Memory (GB)	Reconstruction quality PSNR (↑)	LPIPS (↓)	SSIM (↑)	MAE (↓)	Localization Accuracy Urban (%)	Suburban (%)	Park (%)
ECMU	SegLoc	**0.102**	**15.66**	**0.46**	**0.63**	**0.11**	88.0 / 93.2 / 97.2	83.7 / 89.2 / 93.4	80.5 / 87.5 / 93.1
	PixLoc NV [13]	9.313	21.85	0.28	0.83	0.06	88.3 / 90.4 / 93.7	79.6 / 81.1 / 85.2	61.0 / 62.5 / 69.4
	PixLoc Oracle [13]						92.8 / 95.1 / 98.5	91.9 / 93.4 / 95.8	84.0 / 85.8 / 90.9
	SegLoc full	16.592	19.5	0.31	0.78	0.07	**93.8 / 95.9 / 97.6**	**89.2 / 91.6 / 93.7**	**88.3 / 90.5 / 93.4**

Table A.2. Comparing semantics-based SegLoc with full distributions (full) with feature-based PixLoc on the pose refinement (PR) task in terms of pose accuracy, memory requirements, and privacy of the underlying 3D map representation.

effects the feature space and implicitly effects the segmentation heatmaps. With the exception of refinement on the park slices, the deep metric learning loss ($\mathcal{L}_{MS}$) and the set of consistency losses are complimentary, proving that the model does benefit from multiple training signals. The relative contribution of the consistency losses compared to the deep metric learning loss increases for non-urban areas. This shows that image level information is less sufficient in challenging scenarios.

Clustering variations. First, we investigate the influence of the clustering parameters by training the model with different numbers of initial prototypes K and using two different initial clustering methods, Meanshift [4] and K-means, within the different meta-classes (see Sec. 2 for more details). Additionally, we train a model initialized without clustering these meta-classes, keeping only the 20 most represented ones from the semantic classes in ADE20k [19]. We present these ablative results for the ECMU dataset in Tab A.4. From these results, we observe that increasing the number of clusters, which also increases the representation's dimensionality, does not improve the localization performances. Decreasing the granularity of the segmentation from 100 to 50 degrades the accuracy of pose approximation. Pose refinement seems to mostly benefit from segmentations with only 50 latent classes compared to 100 classes on the park slices. This suggests that the K-means initialization might suffer from over-segmentation which impairs the convergence of the pose refinement method within less discriminative environments. Still, given the pose approximation results, we chose to work with 100 clusters. Not clustering the meta-classes causes the accuracy to drop sharply, letting the model uncover finer details obtained with a finer-grained segmentation seems therefore crucial. Overall, the optimal granularity of the representations seems to largely depend on the scene and the richness of its semantic content.

In Fig. A.6 we show the pseudo targets derived from Eq. (1) (from the main paper) during the training phase ($\mathbf{Q}$) (after the first epoch) and the resulting segmentations after the training (derived from $\mathbf{P}$) for images from the Extended CMU Seasons dataset. Both the K-Means and MeanShift initializations are shown for the models trained with $K = 100$ classes. Segmentations learnt from meanshift initialization show some visually less interpretable clusters with less precise boundaries compared to K-means initialization. Confirmed by the quantitative results in Tab A.4, we decided to use K-means initialization.

Compression and downsampling. In this section, we study the effect of point cloud downsampling. As shown in [3], point cloud downsampling can increase privacy at the price of reduced accuracy. Our method using the full point cloud annotated with cluster indexes has been shown to already ensure a high degree of privacy. As such, while downsampling the point cloud is possible, it would not sig-

	Privacy Preserving	Memory (MB)	scene1	scene2a	scene3	scene4a	scene5	scene6
			Median pose error (cm.) (↓), Median angle error (°) (↓), Recall at 5cm/5° (%) (↑)					
DSAC* [2]	✓	27	12.3/2.06/18.7	7.9/0.9/28.0	13.1/2.34/19.7	**3.7**/0.95/60.8	40.7/6.72/10.6	6.0/1.40/44.3
NBE+SLD [5]	✓	132	6.5/0.9/38.4	7.2/0.68/32.7	4.4/0.91/53.0	3.8/**0.94/66.5**	6.0/0.91/40.0	5.0/0.99/50.5
SegLoc	✓	161	4.8/0.71/44.4	3.4/0.36/54.8	4.6/0.85/39.6	17.0/1.15/20.1	5.9/0.84/40.0	3.9/0.68/33.4
Pixloc [13]	×	28425	**2.6/0.33**/53.5	4.3/0.39/48.0	7.3/1.18/30.4	23.6/1.82/14.8	11.5/1.63/23.7	31.1/3.11/22.5
SegLoc full	?	39968	3.0/0.43./**61.2**	**2.5/0.23/68.1**	**2.1/0.41/61.0**	8.4/0.66/33.56	**3.9/0.57/54.7**	**2.5/0.43/38.8**

Table A.3. Localization results on Indoor6, we run SegLoc with full distributions (full) and compare it with Pixloc.

	Init.	N	urban	suburban	park
PA	mC	20	14.5 / 37.1 / 91.7	5.8 / 20.3 / 82.2	5.9 / 22.3 / 82.4
	KM	50	15.6 / **39.4** / **93.9**	6.3 / 22.1 / 86.2	7.0 / 26.2 / **88.8**
	KM	100	**15.7** / **39.4** / 93.2	**6.4** / **22.5** / **86.6**	**7.4** / **27.1** / 87.6
	KM	256	15.0 / 37.9 / 92.8	6.0 / 21.2 / 84.6	6.7 / 25.4 / 87.8
	MS	50	14.6 / 37.1 / 91.4	5.9 / 21.2 / 81.7	6.3 / 23.9 / 83.1
	MS	100	14.5 / 36.8 / 90.9	6.0 / 21.1 / 80.6	6.3 / 23.4 / 83.5
	MS	256	14.6 / 36.7 / 91.9	6.0 / 21.3 / 84.1	6.3 / 23.6 / 84.5
PR	mC	20	30.9 / 51.9 / 91.4	27.6 / 47.5 / 81.6	22.9 / 41.8 / 80.8
	KM	50	42.0 / 64.4 / **94.4**	**35.9** / **57.7** / 86.5	**34.3** / **56.5** / **88.8**
	KM	100	**44.8** / **64.5** / 93.7	33.9 / 55.7 / **86.9**	29.8 / 52.1 / 87.5
	KM	256	41.1 / 62.5 / 93.2	31.7 / 52.3 / 85.0	28.5 / 50.7 / 87.9
	MS	50	37.0 / 57.9 / 92.0	27.4 / 47.8 / 82.1	24.9 / 45.8 / 83.4
	MS	100	37.9 / 58.1 / 91.4	30.1 / 48.8 / 81.1	26.8 / 46.5 / 83.8
	MS	256	40.9 / 63.1 / 92.5	33.9 / 54.1 / 84.4	29.8 / 50.9 / 84.6

Table A.4. Pose approximation (PA) top-1, and pose refinement (PR) results on the Extended CMU Seasons dataset, when we vary the clustering method and the granularity K of the semantic segmentation for an input size of 480 pixels. mC refers to the meta clusters obtained with the pre-trained model, KM refers to K-Means and MS to MeanShift.

TopK	Ratio(%)	urban	suburban	park
500	10	81.0/89.2/94.3	70.6/79.9/86.1	69.1/78.6/85.9
1000	27	85.7/90.5/94.3	75.3/81.4/85.8	74.1/80.6/85.9
1500	51	85.9/90.4/94.3	75.9/81.2/85.5	75.0/80.6/85.9
All	100	88.0/93.2/97.2	83.7/89.2/93.4	80.5/87.5/93.1

Table A.5. PR with downsampled point clouds on ECMU.

nificantly increase privacy. To assess the effect of compression on the localization accuracy, we subsample the ECMU 3D point clouds by keeping only the top K observations for each database image whose associated 3D points have the highest track length. This is a crude approximation of SOTA algorithms for point cloud compression and is only a simple way to evaluate how our approach behaves with sparser point clouds. After running our pose refinement with these subsampled point clouds, we report results in Tab. A.5 with varying K. We can see that downsampling in general decreases pose accuracy. Despite downsampling 50% of the 3D points, the accuracy drop is within the 3-10% range. While a more advanced compression scheme such as [17] will probably yield a smaller drop in accuracy, it will not further increase privacy.

5. Segmentation-based representations *vs*. descriptor dimensionality reduction

One perspective on our segmentation-based approach is to consider it as a descriptor compression scheme that replaces higher-dimensional descriptors with 1D descriptors (label information). From this perspective, we compare our approach against a simple approach that matches SIFT features extracted from a query image with 3D points associated with SIFT descriptors. The camera pose is then estimated using a P3P solver inside RANSAC with local optimization. We vary the dimensionality of the descriptors using PCA. Table A.6 shows the results of the comparison. As can be seen, state-of-the-art results can be achieved using as few as 6 dimensions. However, using even lower-dimensional descriptors drastically decreases pose accuracy. In contrast, our approach, using 1-dimensional descriptors, still performs well. We observed a similar behavior for PixLoc when PCA-projecting to 3D (albeit without re-training). We were even able to recover images via training an image translation network show- ing that low-dimensional features (3D in this case) are not necessarily privacy preserving. We attribute the difference between our 1D descriptors and the 2D/3D descriptors to learning class labels rather than a metric space.

Model	MB	King's	Old	Shop	St. Mary's
GoMatch [20]	48	0.25/0.64	2.83/8.14	0.48/4.77	3.35/9.94
SegLoc	23	0.24/0.26	0.36/0.52	0.11/0.34	0.17/0.46
PixLoc [13]	3545	0.14/0.24	0.16/0.32	0.05/0.23	0.10/0.34
Sift 16D	16.23	0.07/0.1	0.12/0.23	0.03/0.11	0.03/0.11
Sift 8D	12.62	0.08/012	0.12/0.29	0.03/0.11	0.04/0.14
Sift 6D	11.71	0.09/0.14	0.23/0.39	0.03/0.13	0.06/0.19
Sift 4D	10.82	47.83/49.70	43.31/66.00	2.24/6.42	88.32/116.77

Table A.6. Localization accuracy of down-projected SIFT features on Cambridge Landmarks

6. Convergence of the pose refinement

To visualize the regions of convergence, we measured the errors before and after refinement for each query of ECMU. Fig. A.5 show the percentage of queries with a certain initial pose/rotation error for which refinement converges (from left to right) to within the coarse, medium, or fine threshold. As can be seen, the success of the pose refinement is highly dependent upon the initial pose approximation. If the latter falls within the basin of convergence of the optimization scheme the refinement will converge to some extent. The shape of this basin depends on the complexity of the local underlying scene (geometry and appearance) and also on the visual overlap between the retrieved images and the query (as a smaller visual overlap makes it harder to get enough constraints for refinement). It thus varies greatly between queries.

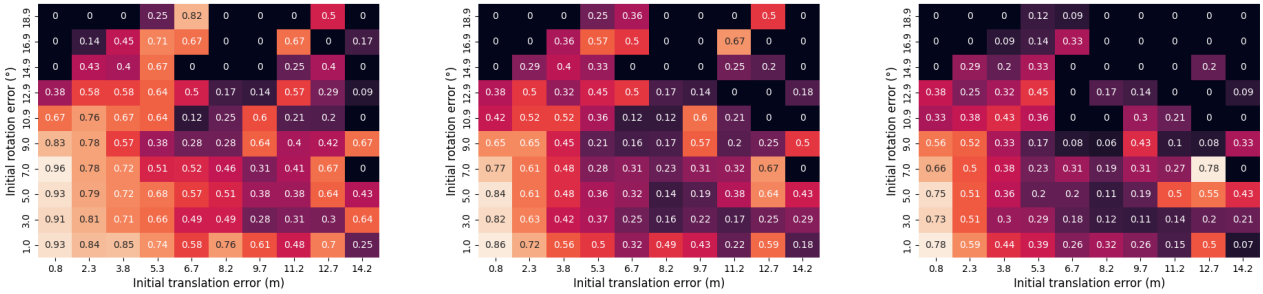

Figure A.5. ECMU, Convergence for coarse, medium and fine thresholds.

7. Qualitative analyses

In Fig. A.7, we provide additional visualizations of SegLoc segmentations. The boundaries between classes are the areas that provide most information for accurate camera pose estimation whereas large uniformly labeled regions are uninformative as small changes in pose might not change the predicted class label. Identifiable borders such as vegetation/sky, building/vegetation, road/sidewalk, floor/furniture are well captured by the models. As such, our approach learns something meaningful. Visually, the clusters maintain some level of explainability and manage to capture fine level details.

References

[1] Hernan Badino, Daniel Huber, and Takeo T. Kanade. Visual Topometric Localization. In *IEEE Intelligent Vehicles Symposium (IVS)*, 2011. 1

[2] Eric Brachmann and Carsten Rother. Visual camera re-localization from RGB and RGB-D images using DSAC. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5847–5865, 2021. 7

[3] Kunal Chelani, Fredrik Kahl, and Torsten Sattler. How Privacy-Preserving Are Line Clouds? Recovering Scene Details From 3D Lines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15668–15678, June 2021. 6

[4] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002. 2, 6

[5] Tien Do, Ondrej Miksik, Joseph DeGol, Hyun Soo Park, and Sudipta N Sinha. Learning To Detect Scene Landmarks for Camera Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11132–11142, 2022. 1, 7

[6] Martin Humenberger, Yohann Cabon, Nicolas Guerin, Julien Morat, Jérôme Revaud, Philippe Rerole, Noé Pion, Cesar de Souza, Vincent Leroy, and Gabriela Csurka. Robust Image Retrieval-based Visual Localization using Kapture. arXiv:2007.13867, 2020. 1

[7] Revaud Jerome, Philippe Weinzaepfel, César De Souza, and Martin Humenberger. R2D2: Reliable and Repeatable Detectors and Descriptors. In *NeurIPS*, 2019. 1

[8] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015. 2

[9] Mans Larsson, Erik Stenborg, Lars Hammarstrand, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. A Cross-Season Correspondence Dataset for Robust Semantic Segmentation. In *CVPR*, 2019. 1

[10] Mans Larsson, Erik Stenborg, Carl Toft, Lars Hammarstrand, Torsten Sattler, and Fredrik Kahl. Fine-Grained Segmentation Networks: Self-Supervised Segmentation for Improved Long-Term Visual Localization. In *ICCV*, 2019. 2

[11] Francesco Pittaluga, Sanjeev J Koppal, Sing Bing Kang, and Sudipta N Sinha. Revealing scenes by inverting structure from motion reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 145–154, 2019. 4

[12] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision Transformers for Dense Prediction. In *ICCV*, 2021. 2

[13] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans 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. 6, 7

[14] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion Revisited. In *CVPR*, 2016. 1

[15] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise View

Selection for Unstructured Multi-View Stereo. In *ECCV*, 2016. 1

[16] Carl Toft, Will Maddern, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, et al. Long-term visual localization revisited. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1

[17] Luwei Yang, Rakesh Shrestha, Wenbo Li, Shuaicheng Liu, Guofeng Zhang, Zhaopeng Cui, and Ping Tan. Scenesqueezer: Learning to compress scene for camera relocalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8259–8268, 2022. 7

[18] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 5

[19] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic Understanding of Scenes through the ade20k Dataset. *International Journal of Computer Vision (IJCV)*, 127:302–321, 2019. 2, 6

[20] Qunjie Zhou, Sergio Agostinho, Aljosa Osep, and Laura Leal-Taixe. Is Geometry Enough for Matching in Visual Localization? *arXiv preprint arXiv:2203.12979*, 2022. 7

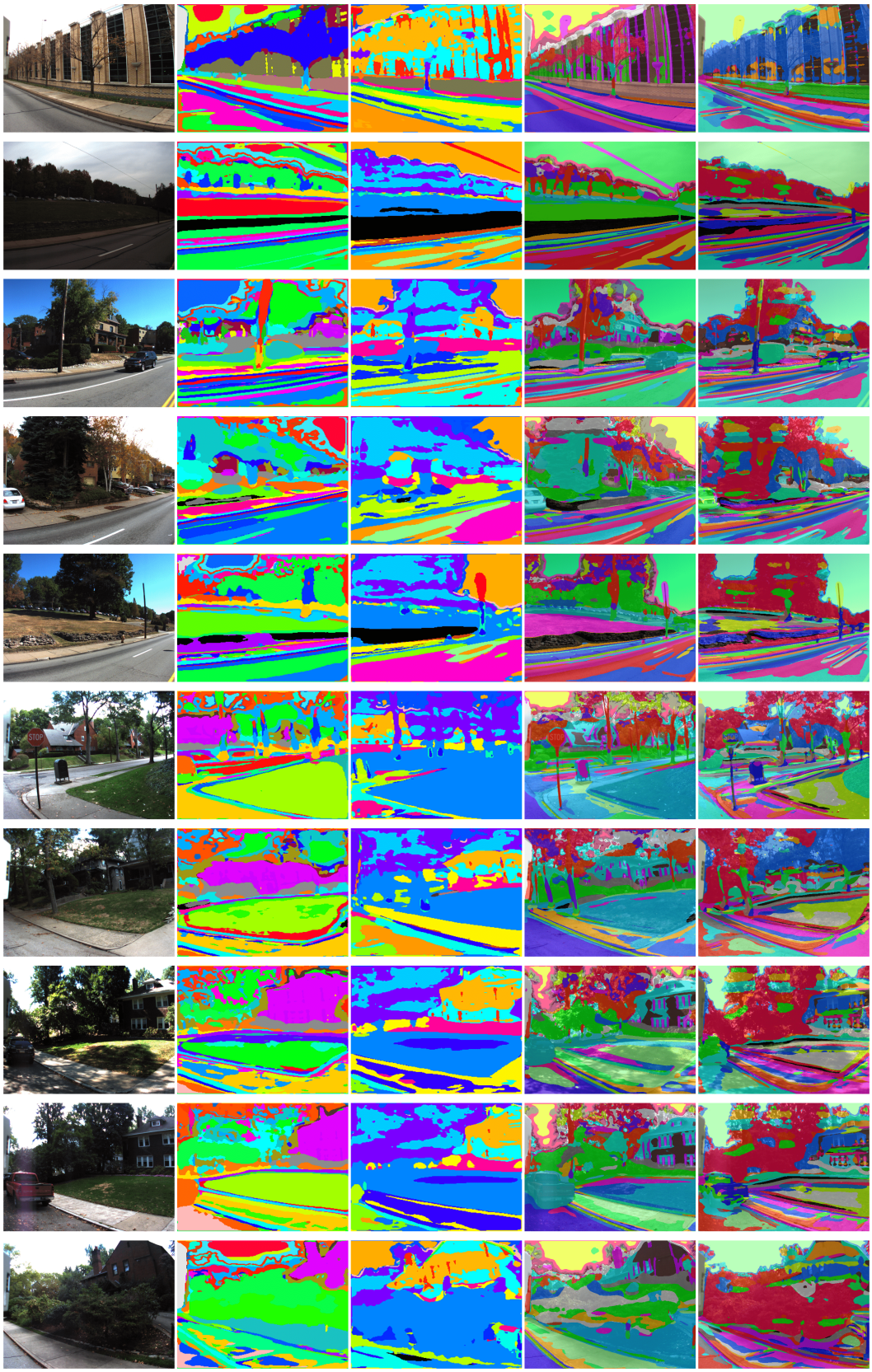

Figure A.6. Segmentation results for the Extended CMU Seasons dataset. From left to right: original image, $\mathbf{Q}$ K-means initialization, $\mathbf{Q}$ MeanShift initialization, $\mathbf{P}$ K-means initialization, $\mathbf{P}$ MeanShift initialization.

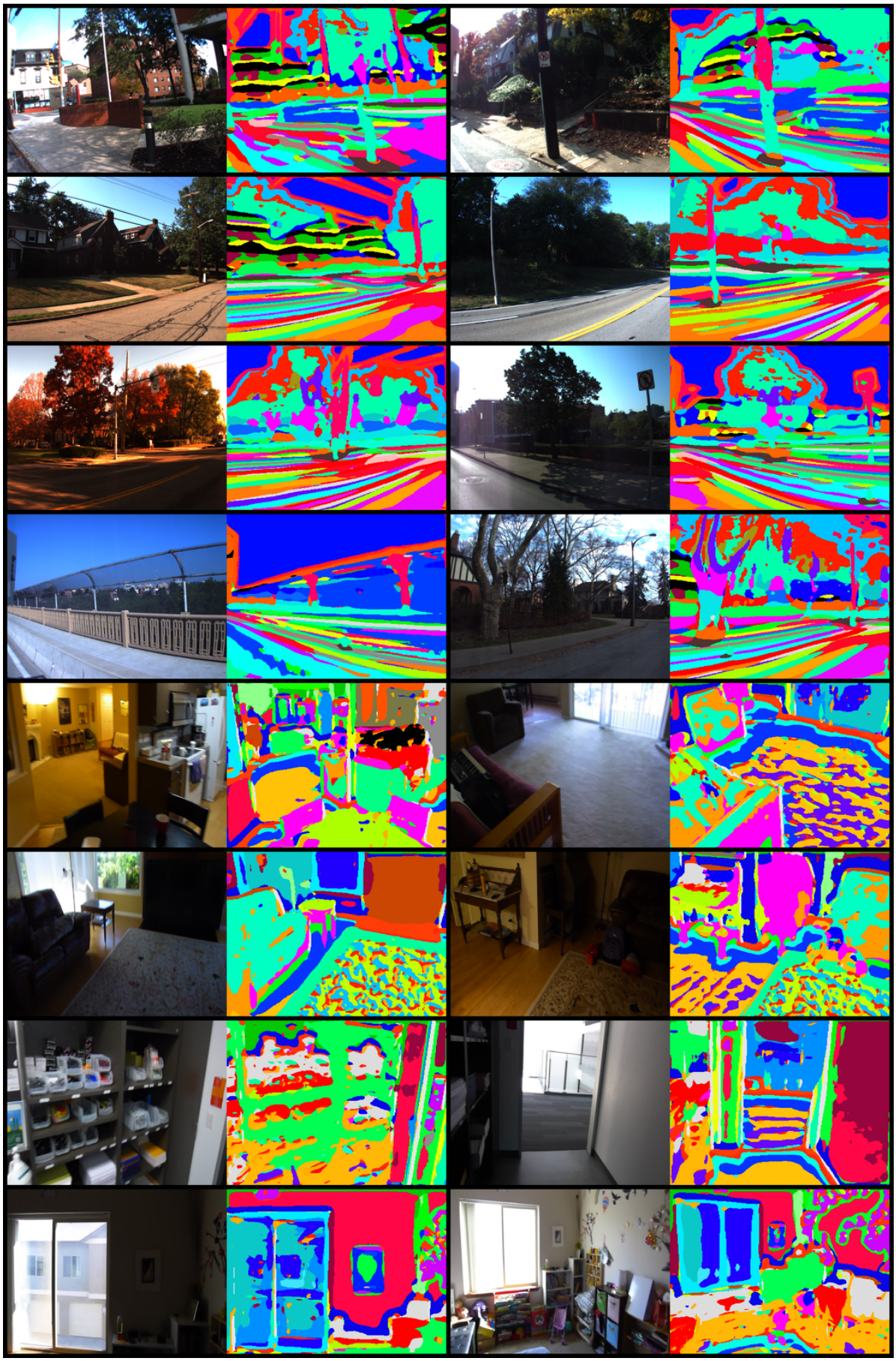

Figure A.7. SegLoc segmentations for the Extended CMU Seasons and Indoor6 datasets