Hierarchical Scene Coordinate Classification and Regression for Visual Localization

Xiaotian Li[1] Shuzhe Wang[1] Yi Zhao[1] Jakob Verbeek[2*] Juho Kannala[1]
[1]Aalto University [2]Facebook AI Reseach

Abstract

Visual localization is critical to many applications in computer vision and robotics. To address single-image RGB localization, state-of-the-art feature-based methods match local descriptors between a query image and a pre-built 3D model. Recently, deep neural networks have been exploited to regress the mapping between raw pixels and 3D coordinates in the scene, and thus the matching is implicitly performed by the forward pass through the network. However, in a large and ambiguous environment, learning such a regression task directly can be difficult for a single network. In this work, we present a new hierarchical scene coordinate network to predict pixel scene coordinates in a coarse-to-fine manner from a single RGB image. The network consists of a series of output layers, each of them conditioned on the previous ones. The final output layer predicts the 3D coordinates and the others produce progressively finer discrete location labels. The proposed method outperforms the baseline regression-only network and allows us to train compact models which scale robustly to large environments. It sets a new state-of-the-art for single-image RGB localization performance on the 7-Scenes, 12-Scenes, Cambridge Landmarks datasets, and three combined scenes. Moreover, for large-scale outdoor localization on the Aachen Day-Night dataset, we present a hybrid approach which outperforms existing scene coordinate regression methods, and reduces significantly the performance gap w.r.t. explicit feature matching methods.[1]

1. Introduction

Visual localization aims at estimating precise six degree-of-freedom (6-DoF) camera pose with respect to a known environment. It is a fundamental component of many intelligent autonomous systems and applications in computer vision and robotics, *e.g.*, augmented reality, autonomous driving, or camera-based indoor localization for personal assistants. Commonly used visual localization methods rely on matching local visual descriptors [43, 44]. Correspondences are typically established between 2D interest points in the query and 3D points in the pre-built structure-from-motion model [49, 50] with nearest neighbor search, and the 6-DoF camera pose of the query can then be computed from the correspondences.

Instead of explicitly establishing 2D-3D correspondences via matching descriptors, scene coordinate regression methods directly regress 3D scene coordinates from an image [3, 5, 8, 51]. In this way, correspondences between 2D points in the image and 3D points in the scene can be obtained densely without feature detection and description, and explicit matching. In addition, no descriptor database is required at test time since the model weights encode the scene representation implicitly. It was experimentally shown that recent CNN-based scene coordinate regression methods achieve better localization performance on small-scale datasets compared to the state-of-the-art feature-based methods [5]. The high accuracy and the compact representation of a dense scene model make scene coordinate regression approach an interesting alternative to the classic feature-based approach.

However, most existing scene coordinate regression methods can only be adopted on small-scale scenes. Typically, scene coordinate regression networks are designed to have a limited receptive field [3, 5], *i.e.* only a small local image patch is considered for each scene coordinate prediction. This allows the network to generalize well from limited training data, since local patch appearance is more stable *w.r.t.* viewpoint change. On the other hand, a limited receptive field size can lead to ambiguous patterns in the scene, especially in large-scale environments, caused by visual similarity between local image patches. Due to these ambiguities, it is harder for the network to accurately model the regression problem, resulting in inferior performance at test time. Using larger receptive field sizes, up to the full image, to regress the coordinates can mitigate the issues caused by ambiguities. This, however, has been shown to be prone to overfitting the larger input patterns in the case of limited training data, even if data augmentation alleviates

[*] Work done while JV was at INRIA.

[1] Code and materials available at https://aaltovision.github.io/hscnet.

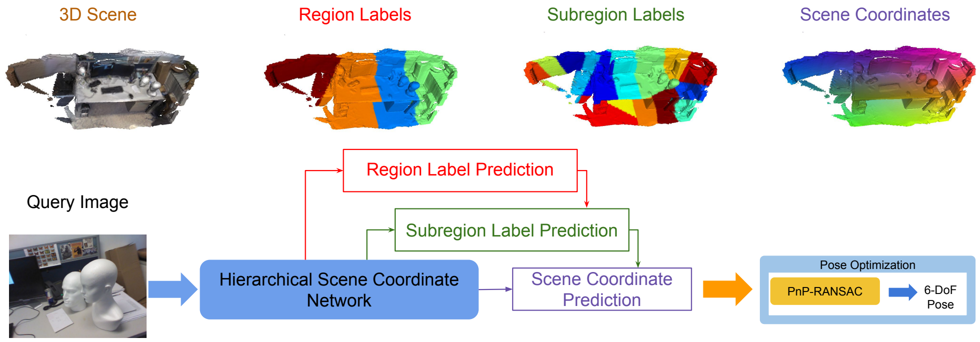

Figure 1. Overview of our single-image RGB localization approach based on hierarchical scene coordinate prediction, here using 3 levels.

this problem to some extent [28].

In contrast, in this work, we overcome the ambiguities due to small receptive fields by conditioning on discrete location labels around each pixel. During training, the labels are obtained by a coarse quantization of the ground-truth 3D coordinates. At test time, the location labels for each pixel are obtained using dense classification networks, which can more easily deal with the location ambiguity since they are trained using the cross-entropy classification loss which permits a multi-modal prediction in 3D space. Our model allows for several classification layers, using progressively finer location labels, obtained through hierarchical clustering of the ground-truth 3D point cloud data. Our hierarchical coarse-to-fine architecture is implemented using conditioning layers that are related to the FiLM architecture [37], resulting in a compact model. See Fig. 1 for a schematic overview of our approach.

We validate our approach by comparing it to a regression-only network, which lacks the hierarchical coarse-to-fine structure. We present results on three datasets used in previous works: 7-Scenes [51], 12-Scenes [57], and Cambridge Landmarks [24]. Our approach shows consistently better performance and achieves state-of-the-art results for single-image RGB localization. Moreover, by compiling the 7-Scenes and 12-Scenes datasets into single large scenes, and using the Aachen Day-Night dataset [45, 47], we show that our approach scales more robustly to larger environments.

In summary, our contributions are as follows:

- We introduce a new hierarchical coarse-to-fine conditioning architecture for scene coordinate prediction, which improves the performance and scalability over a baseline regression-only network.
- We show that our novel approach achieves state-of-the-art results for single-image RGB localization on three benchmark datasets and it allows us to train single compact models which scale robustly to large scenes.
- For large-scale outdoor localization, we present a hybrid approach built upon our network, which reduces significantly the gap to feature-based methods.

2. Related Work

Visual localization. Visual localization aims at predicting 6-DoF camera pose for a given query image. To obtain precise 6-DoF camera pose, visual localization methods are typically structure-based, *i.e.* they rely on 2D-3D correspondences between 2D image positions and 3D scene coordinates. With the established 2D-3D correspondences, a RANSAC [20] optimization scheme is responsible for producing the final pose estimation. The correspondences are typically obtained by matching local features such as SIFT [30], and many matching and filtering techniques have been proposed, which enable efficient and robust city-scale localization [15, 26, 35, 44, 53, 55].

Image retrieval can also be used for visual localization [1]. The pose of the query image can be directly approximated by the most similar retrieved database image. Since compact image-level descriptors are used for matching, image retrieval methods can scale to very large environments. The retrieval methods can be combined with structure-based methods [41, 42, 46, 54, 61] or relative pose estimation [2, 18, 27] to predict precise poses. Typically, the retrieval step helps restrict the search space, leading to faster and more accurate localization.

In recent years, learning-based localization approaches have been explored. One popular direction is to replace the entire localization pipeline with a single neural network. PoseNet [24] and its variants [9, 22, 23, 32, 59] directly regress the camera pose from a query image. Recently, however, it was demonstrated that direct pose regression yields results more similar to pose approximation via image retrieval than to accurate pose estimation via 3D structure [48]. Therefore, these methods are still outperformed by structure-based methods. By fusing estimated pose information from the previous frame, [38, 56] achieve better performance, but require sequences of images rather than single images.

Scene coordinate regression. Instead of learning the entire pipeline, scene coordinate regression methods learn the first stage of the pipeline in the structure-based approaches. Namely, either a random forest [4, 13, 14, 21, 31, 33, 34, 51, 58] or a neural network [3, 5, 6, 7, 8, 10, 11, 12, 28, 29, 31] is trained to directly predict 3D scene coordinates for the pixels and thus the 2D-3D correspondences are established. These methods do not explicitly rely on feature detection, description and matching, and are able to provide correspondences densely. They are more accurate than traditional feature-based methods at small and medium scale, but usually do not scale well to larger scenes [5, 6]. In order to generalize well to novel viewpoints, these methods typically rely on only local image patches to produce the scene coordinate predictions. However, this may introduce ambiguities due to similar local appearances, especially when the scale of the scene is large. To resolve local appearance ambiguities, we introduce element-wise conditioning layers to modulate the intermediate feature maps of the network using coarse discrete location information. We show this leads to better localization performance, and we can robustly scale to larger environments.

Joint classification-regression. Joint classification-regression frameworks have been proved effective in solving various vision tasks. For example, [39, 40] proposed a classification-regression approach for human pose estimation from single images. In [4], a joint classification-regression forest is trained to predict scene identifiers and scene coordinates. In [60], a CNN is used to detect and segment a predefined set of planar Objects-of-Interest (OOIs), and then, to regress dense matches to their reference images. In [10], scene coordinate regression is formulated as two separate tasks of object instance recognition and local coordinate regression. In [6], multiple scene coordinate regression networks are trained as a mixture of experts along with a gating network which assesses the relevance of each expert for a given input, and the final pose estimate is obtained using a novel RANSAC framework, *i.e.*, Expert Sample Consensus (ESAC). In contrast to existing approaches, in our work, we use spatially dense discrete location labels defined for all pixels, and propose FiLM-like [37] conditioning layers to propagate information in the hierarchy. We show that our novel framework allows us to achieve high localization accuracy with one single compact model.

3. Hierarchical Scene Coordinate Prediction

We now describe our coarse-to-fine hierarchical scene coordinate prediction approach. Note that we address single-image RGB localization, as in *e.g.* [5, 6, 7, 29], rather than using RGB-D images [12, 13, 14, 21, 34, 51, 58], or image sequences [38, 56].

Hierarchical joint learning framework. To define hierarchical discrete location labels, we hierarchically partition the ground-truth 3D point cloud data. This step can be done, *e.g.*, with k-means. In this way, in addition to the ground-truth 3D scene coordinates, each pixel in a training image is also associated with a number of labels, from coarse to fine, obtained at different levels of the clustering hierarchy. Then, for each level, our network has a corresponding classification layer which for all pixels predicts the discrete location labels at that level. Besides the classification layers, we include a final regression layer to predict the continuous 3D scene coordinates for the pixels, generating putative 2D-3D matches. To propagate the coarse location information to inform the predictions at finer levels, we introduce conditioning layers before each classification/regression layer. Note that we condition on the ground truth label maps during training, and condition on the predicted label maps at test time.

Since the predictions in each classification layer are conditioned on all preceding label maps, at each particular classification layer, it suffices to predict the label branch at that level. For example, for a three-level classification hierarchy, with branching factor k, we classify across only k labels at each level. Similar to [10], instead of directly regressing the absolute coordinates, we regress the relative positions to the cluster centers in 3D space at the finest level. This accelerates convergence of network training [10]. Note that this hierarchical scene coordinate learning framework also allows a classification-only variant. That is, if we have fine enough location labels before the regression layer, we can simply use the cluster centers as the scene coordinates predictions without performing a final regression step.

We design the network to be global-to-local, which means that finer output layers have smaller receptive fields in the input image. This allows the network to use more global information at coarser levels, while conditioning on location labels to disambiguate the local appearances at finer levels. Note that at test time, the receptive fields of the finer output layers are also large, as they depend on the discrete location labels which are predicted from the input at test time, rather than fixed as during training.

Conditioning layers. To make use of the discrete location label information predicted by the network at coarser levels, these predictions should be fed back to the finer levels. Inspired by the Feature-wise Linear Modulation (FiLM) conditioning method [37], we introduce conditioning layers just before each of the output layers. A conditioning parameter generator takes the predicted label map ℓ as input, outputs a set of scaling and shifting parameters $\gamma(\ell)$ and $\beta(\ell)$, and these parameters are fed into the conditioning layer to apply linear transformation to the input feature map. Unlike FiLM layers, however, which perform

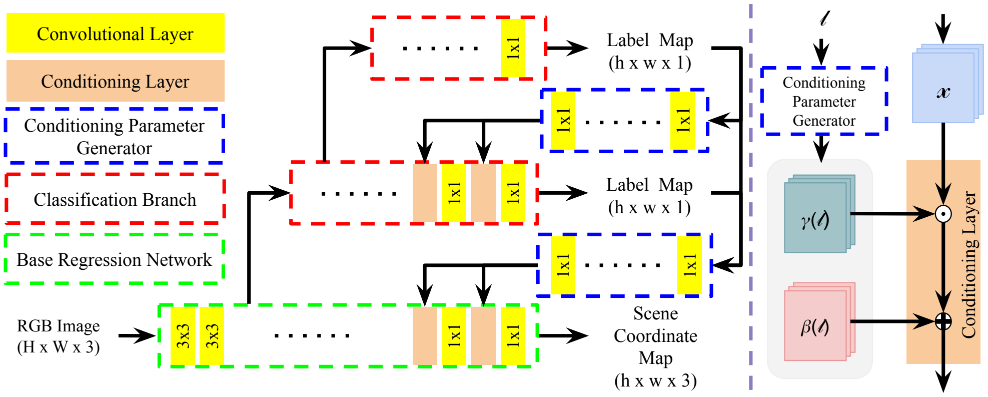

Figure 2. Left: Architecture of our hierarchical scene coordinate network (3-level). Right: Architecture of the conditioning layer.

the same channel-wise modulation across the entire feature map, our conditioning layers perform a linear modulation per spatial position, *i.e.*, element-wise multiplication and addition as shown in Fig. 2 (right). Therefore, instead of vectors, the output parameters $\gamma(\ell)$ and $\beta(\ell)$ from a generator are feature maps of the same (height, width, channel) dimensions as the input feature map of the corresponding conditioning layer. More formally, given the input feature map x, the scaling and shifting parameters $\gamma(\ell)$ and $\beta(\ell)$, the linear modulation can be written as:

$$f(x, \ell) = \gamma(\ell) \odot x + \beta(\ell), \tag{1}$$

where $\odot$ denotes the Hadamard product. In addition, the generators consist of only 1×1 convolutional layers so that each pixel is conditioned on its own location labels. We use an ELU non-linearity [16] after the feature modulation.

Network architecture. In our main experiments we use 3-level hierarchy for all the datasets, *i.e.* our network has two classification output layers and one regression output layer. The overall architecture of this network is shown in Fig. 2 (left). The first classification branch predicts the coarse location labels, and the second one predicts the fine labels. We use strided convolution, upconvolution and dilated convolution for the two classification branches to enlarge the size of the receptive field, while preserving the output resolution. All the layers after the conditioning layers have kernel size of 1×1 such that the label conditioning is applied locally. More details on the architecture are provided in the supplementary material.

Loss function. Our network predicts location labels and regresses scene coordinates at the same time. Therefore, we need both a regression loss and a classification loss during training. For the regression task, we minimize the Euclidean distance between predicted scene coordinates $\hat{y}$ and ground truth scene coordinates y,

$$\mathcal{L}_r = \sum_i \|y_i - \hat{y}_i\|_2, \tag{2}$$

where i ranges over the pixels in the image. For the classification task, we use cross-entropy loss at each level, *i.e.*

$$\mathcal{L}_c^j = -\sum_i \left(\ell_i^j\right)^\top \log \hat{\ell}_i^j, \tag{3}$$

where ℓ_i^j denotes the one-hot coding of the ground-truth label of pixel i at level j, and $\hat{\ell}_i^j$ denotes the vector of predicted label probabilities for the same pixel, and the logarithm is applied element-wise. In the case of 3-level hierarchy, the final loss function is given by

$$\mathcal{L} = w_1 \mathcal{L}_c^1 + w_2 \mathcal{L}_c^2 + w_3 \mathcal{L}_r, \tag{4}$$

where w_1, w_2, w_3 are weights for the loss terms. We found that the accuracy of the final regression prediction is crucial to localization performance, and thus a large value should be set for the regression loss. Details on the weights and training procedure are provided in the supplementary material. Note that, as mentioned before, our hierarchical joint learning framework also allows a classification-only variant, by using a finer label hierarchy.

4. Experimental Evaluation

In this section, we present our experimental setup and evaluation results on standard visual localization datasets.

4.1. Datasets and Experimental Setup

We use four standard benchmark datasets for our experiments. The **7-Scenes** (7S) [51] dataset is a widely used RGB-D dataset that contains seven indoor scenes. RGB-D image sequences of the scenes are recorded by a KinectV1. Ground truth poses and dense 3D models are also provided. **12-Scenes** (12S) [57] is another indoor RGB-D dataset. It is composed of twelve rooms captured with a Structure.io depth sensor and an iPad color camera, and ground truth poses are provided along with the RGB-D images. The recorded environments are significantly larger than those

7-Scenes	DSAC++ [5]		AS [44]		Inloc [54]		Regression-only		Ours	
——	**Acc.**	**Med. Err.**	**Acc.**	**Med. Err.**	**Acc.**	**Med. Err.**	**Acc.**	**Med. Err.**	**Acc.**	**Med. Err.**
Chess	97.1	**0.02**, **0.5**	-	0.04, 2.0	-	0.03, 1.1	95.4	**0.02**, 0.7	**97.5**	**0.02**, 0.7
Fire	89.6	**0.02**, **0.9**	-	0.03, 1.5	-	0.03, 1.1	94.9	**0.02**, **0.9**	**96.7**	**0.02**, **0.9**
Heads	92.4	**0.01**, **0.8**	-	0.02, 1.5	-	0.02, 1.2	97.1	**0.01**, **0.8**	**100**	**0.01**, 0.9
Office	**86.6**	**0.03**, **0.7**	-	0.09, 3.6	-	**0.03**, 1.1	81.4	**0.03**, 0.9	86.5	**0.03**, 0.8
Pumpkin	59.0	**0.04**, 1.1	-	0.08, 3.1	-	0.05, 1.6	58.0	**0.04**, 1.1	**59.9**	**0.04**, **1.0**
Kitchen	**66.6**	**0.04**, **1.1**	-	0.07, 3.4	-	**0.04**, 1.3	56.5	0.05, 1.4	65.5	**0.04**, 1.2
Stairs	29.3	0.09, 2.6	-	**0.03**, 2.2	-	0.09, 2.5	68.1	0.04, 1.0	**87.5**	**0.03**, **0.8**
Average	74.4	0.04, 1.1	-	0.05, 2.5	-	0.04, 1.4	78.8	**0.03**, 1.0	**84.8**	**0.03**, **0.9**
Complete	76.1		-		-		74.7		**80.5**	

Cambridge	DSAC++ [5]	AS [44]	NG-RANSAC [7]	Regression-only	Ours
Great Court	0.40, **0.2**	-	0.35, -	1.25, 0.6	**0.28**, **0.2**
K. College	0.18, **0.3**	0.42, 0.6	**0.13**, -	0.21, **0.3**	0.18, **0.3**
Old Hospital	0.20, **0.3**	0.44, 1.0	0.22, -	0.21, **0.3**	**0.19**, **0.3**
Shop Facade	**0.06**, **0.3**	0.12, 0.4	**0.06**, -	**0.06**, **0.3**	**0.06**, **0.3**
St M. Church	0.13, 0.4	0.19, 0.5	0.10, -	0.16, 0.5	**0.09**, **0.3**
Average	0.19, **0.3**	0.29, 0.6	0.17, -	0.38, 0.4	**0.16**, **0.3**

12-Scenes	DSAC++ [5]		Regression-only		Ours	
——	**Acc.**	**Med. Err.**	**Acc.**	**Med. Err.**	**Acc.**	**Med. Err.**
Kitchen-1	**100**	-	**100**	**0.008**, **0.4**	**100**	**0.008**, **0.4**
Living-1	**100**	-	**100**	**0.011**, **0.4**	**100**	**0.011**, **0.4**
Bed	99.5	-	**100**	0.013, 0.6	**100**	**0.009**, **0.4**
Kitchen-2	99.5	-	**100**	0.008, 0.4	**100**	**0.007**, **0.3**
Living-2	**100**	-	**100**	0.014, 0.6	**100**	**0.010**, **0.4**
Luke	95.5	-	93.8	0.020, 0.9	**96.3**	**0.012**, **0.5**
Gates 362	**100**	-	**100**	0.011, 0.5	**100**	**0.010**, **0.4**
Gates 381	96.8	-	98.8	0.016, 0.7	**99.1**	**0.012**, **0.6**
Lounge	95.1	-	99.4	0.015, **0.5**	**100**	**0.014**, **0.5**
Manolis	96.4	-	97.2	0.014, 0.7	**100**	**0.011**, **0.5**
Floor5a	83.7	-	97.0	0.016, 0.7	**98.8**	**0.012**, **0.5**
Floor 5b	95.0	-	93.3	0.019, 0.6	**97.3**	**0.015**, **0.5**
Average	96.8	-	98.3	0.014, 0.6	**99.3**	**0.011**, **0.5**
Complete	96.4		97.9		**99.1**	

Table 1. The median errors (m, °) for 7-Scenes, 12-Scenes and Cambridge, and the percentages of accurately localized test images (error < 5 cm, 5°) for 7-Scenes and 12-Scenes. "Complete" refers to the percentage among all test images of all scenes.

in 7-Scenes. **Cambridge Landmarks** [24] is an outdoor RGB visual localization dataset. It consists of RGB images of six scenes captured using a Google LG Nexus 5 smartphone. Ground truth poses and sparse 3D reconstructions generated with structure from motion are also provided. In addition to these three datasets, we synthesize three large-scale indoor scenes based on 7-Scenes and 12-Scenes by placing all seven, twelve or nineteen individual scenes, into a single coordinate system similar to [6]. These large integrated datasets are denoted by **i7-Scenes** (i7S), **i12-Scenes** (i12S), **i19-Scenes** (i19S), respectively. Finally, we evaluate our method on the **Aachen Day-Night** dataset [45, 47] which is very challenging for scene coordinate regression methods due to the scale and sparsity of the 3D model. In addition, it contains a set of challenging night time queries, but there is no night time training data. In the following, we present the main setup for experiments on all the datasets except Aachen. See supplementary for details on Aachen.

Ground truth scene coordinates can be either obtained from the known poses and depth maps or rendered using a 3D model. To generate the ground truth location labels, we run hierarchical k-means clustering on dense point cloud models. For all the individual scenes used in the main experiments, unless stated otherwise, we use two-level hierarchical k-means with the branching factor set to 25 for both levels. For the three combined scenes, i7-Scenes, i12-Scenes, and i19-Scenes, we simply combine the label trees at the first level. That is, *e.g*., for the i7-Scenes, there are 175 branches in total at the first level.

We use the same VGG-style [52] architecture as DSAC++ [5] as the base regression network for our method, except we use ELU activation [16] instead of ReLU [36]. This is because we found that the plain regression network is easier to train with ReLU, while our network which has the additional conditioning layers and classification branches works better with ELU. The regression layer, the second and first classification layer have a receptive field size of 73×73, 185×185, and 409×409 pixels, respectively, in the input image. To show the advantage of the proposed architecture, we also evaluate the localization performance of the same regression-only network used in DSAC++ [5], but here trained with the Euclidean loss term only. Note that in [5], two additional training steps are proposed and the entire localization pipeline is optimized end-to-end, which can further improve the accuracy. Potentially, our network can also benefit from the DSAC++ framework, but it is beyond the scope of the current paper. Unless specified otherwise, we perform affine data augmentation with additive brightness changes during training. We also report the results obtained without data augmentation in Sec. 4.4. For pose estimation, we follow [5], and use the same PnP-RANSAC algorithm with the same hyperparameter settings. Further details about the architecture, training and other settings can be found in the supplementary material.

4.2. Results on 7-Scenes, 12-Scenes and Cambridge

To evaluate our hierarchical joint learning architecture, we first compare it with the state-of-the-art methods as well as a regression-only baseline on the 7-Scenes, the 12-Scenes, and the Cambridge Landmarks datasets. For the Cambridge Landmarks, we report median pose accuracy as in the previous works. Following [5, 7, 29], we do not include the Street scene, since the dense 3D reconstruction of this scene has rather poor quality that hampers performance. For the 7-Scenes and the 12-Scenes, we also report the percentage of the test images with error below 5 cm and 5°, which is used as the main evaluation metric for both datasets and gives more information about the localization performance. Scene coordinate regression methods are currently the best performing single-image RGB methods on these three small/medium scale datasets [5, 7]. We also compare to a state-of-the-art feature-based method, *i.e*. Active Search [44] and an indoor localization method which exploits dense correspondences [54]. Note that, in general, methods that exploit additional depth information [12, 13] or sequences of images [38, 56] can provide better localization performance. However, the additional required information also restricts the scenarios in which they can be applied. We do not compare to those methods in this work, since they are not directly comparable to our results pro-

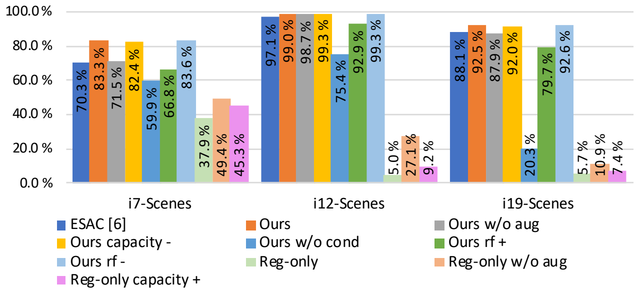

Figure 3. Average pose accuracy on the combined scenes. Results for ESAC taken from [6]. Our method consistently outperforms the regression-only baseline by a large margin and achieves better performance compared to ESAC.

duced in the single-image RGB localization setting.

The results are reported in Table 1. Numbers for the competing methods are taken from the corresponding papers. Overall, our approach yields excellent results. Compared to the regression-only baseline, our approach provides consistently better localization performance on all the scenes across the three datasets. During training, we also observed consistently lower regression training error compared to the regression-only baseline, underlining the ability of the discrete location labels to disambiguate the local appearances. Our approach also achieves overall better results compared to the current state-of-the-art methods DSAC++ [5] on all three datasets, and NG-RANSAC [7] on the Cambridge Landmarks (the latter does not report results on the 7-Scenes and 12-Scenes datasets).

In Table 1 we trained our networks and the regression-only baseline with data augmentation, while DSAC++ and NG-RANSAC did not use data augmentation. In Sec. 4.4, we show that even without data augmentation, our method still achieves comparable or better performance compared to DSAC++ and NG-RANSAC. Moreover, in DSAC++ and NG-RANSAC, more advanced training steps and RANSAC schemes are proposed to improve the accuracy of the plain regression network and to optimize the entire pipeline, while in this work we focus on the scene coordinate network itself and we show that improvements on this single component can already improve the localization performance beyond the state-of-the-art. Note that DSAC++ and NG-RANSAC are complementary to our approach, and their combination could be explored in future work.

4.3. Results on Combined Scenes

The individual scenes from the previous datasets all have very limited physical extent. As in [6], to go beyond such small environments, we use the combined scenes, *i.e*. the i7-Scenes, i12-Scenes, and the i19-Scenes, as described in Sec. 4.1. We mainly compare to the regression-only baseline and ESAC [6] on the three combined scenes. To the best of our knowledge, ESAC is currently the only scene coordinate regression method that scales well to the combined scenes. The results are reported in Fig. 3.

Reg-only	Ours	Ours capacity-	ESAC (i7S) [6]	ESAC (i12S) [6]	ESAC (i19S) [6]
104MB	165MB	73MB	7×28MB	12×28MB	19×28MB

Table 2. Model size comparison. Our method can scale robustly to large environments with a compact model.

We see that the localization performance of the regression baseline (*Reg-only*) decreases dramatically when trained on the combined scenes compared to trained and tested on each of the scenes individually, *c.f*. Table 1. Its performance drops more drastically as the scene grows larger. Our method is much more robust to the increase in the environment size, and significantly outperforms the baseline. This underlines the importance of our hierarchical learning framework when the environment is large and potentially contains more ambiguities. Our method also outperforms ESAC which uses an ensemble of networks, where each network specializes in a local part of the environment [6]. ESAC requires to train and store multiple networks, whereas our approach requires only a single model.

Note that for ESAC the authors did not use data augmentation. When we train our method without data augmentation (*Ours w/o aug*), we still outperform ESAC on i7-Scenes and i12-Scenes, and obtain a slightly lower but comparable accuracy on i19-Scenes (87.9% *vs*. 88.1%). Note that ESAC and our approach are complementary, and their combination could be explored in future work.

4.4. Detailed Analysis

Network capacity. Compared to the regression-only baseline, our network has extra layers for the conditioning generators and classification branches, and thus has an increased number of parameters. Therefore, for fair comparison, we add more channels to the regression-only baseline to compensate the increased number of parameters in our model. On 7-Scenes, the average accuracy of the regression baseline increased from 78.8% to 80.4%. On the combined scenes, as shown in Fig. 3, we observe larger improvement in performance (denoted by *Reg-only capacity*+ in Fig. 3). However, even with increased capacity, the regression-only baseline still lags far behind our method, especially on the combined scenes.

We also experimented with reducing the size of the backbone regression network, which accounts for most of the model parameters. We add more conditioning layers early in the network, while using less shared layers between the regression and classification branches. We denote the resulting network by *Ours capacity*-, see supplementary for details. In Table 2, we compare the model size of our network to the regression baseline and ESAC on the combined

	7-Scenes	12-Scenes	Cambridge
Reg-only w/o aug	70.9%	97.5%	0.38m, 0.4°
Ours w/o aug	75.5%	99.4%	0.18m, 0.3°
DSAC++ [5]	74.4%	96.8%	0.19m, 0.3°
NG-RANSAC [7]	-	-	0.17m, -

Table 3. Average pose accuracy/median error on the 7-Scenes, 12-Scenes and Cambridge datasets of our method and the regression-only baseline without data augmentation.

scenes. We see in Fig. 3 and Table 2 that this allows us to reduce our model size by more than a factor of two, while incurring a loss in accuracy below one percentage point. Compared to ESAC on the i19-Scenes dataset, our compressed model is more than seven times more compact. Note that since we perform regression locally, the k-means cluster centers also need to be stored. Since for each individual scene there are only 625 clusters, the storage space needed for the cluster centers is negligible ($<$ 1MB).

Using global information. Using global information directly to regress scene coordinates has been explored in [28]. However, even with data augmentation, large input patterns remain sensitive to viewpoint changes, leading to inferior performance at test time compared to using local patches [5]. We validate this by using the same regression network, but now with dilated convolution such that the receptive field size is much larger (409×409). We find that in general directly using global context helps the training loss decrease faster. This might have a positive effect on complex scenes (39.3% with dilated convolution *vs*. 37.9% without it on i7-Scenes). For less demanding scenes, however, the network usually gives worse results (59.2% *vs*. 78.8% on 7-Scenes) due to decreased viewpoint invariance. Meanwhile, our network is able to use the global information in a more robust way, *i.e*., indirectly through discrete location labels.

We also created two variants of our network with small (73×73) and large (409×409) receptive field across all levels, denoted by *Ours rf-* and *Ours rf+* respectively in Fig. 3. As expected, increasing the receptive field size at all levels harms the performance, as shown in Fig. 3. Interestingly, the model with small receptive field even performs sightly better on the combined scenes. This indicates that the local ambiguities can be handled well by the hierarchical coarse-to-fine conditioning mechanism.

Data augmentation. We apply affine transformations to the images with additive brightness changes as data augmentation during training. In general, this improves the generalization capability of the network and makes it more robust to lighting and viewpoint changes. According to Table 1, Table 3 and Fig. 3, data augmentation consistently improves the localization performance of our method, except on the 12-Scenes dataset; in 12S, the training and test trajectories are close, and there are no significant viewpoint changes between training and test frames [13]. Data augmentation, however, can also increase the appearance ambiguity of the training data and make the network training more difficult. This happens to the baseline regression-only network: Although data augmentation helps it on the small-scale scenes, on the Cambridge and the combined scenes, data augmentation has no positive effects and even harms the performance. Note that without data augmentation, our method still provides results that are better than or on par with the state-of-the-arts, see Table 3 and Fig. 3.

7S	9×9	49×49	10×100×100	10×100×100×100	625	25×25
	82.9%	85.0%	85.9%	85.5%	85.3%	84.8%
i7S	63×9	343×49	70×100×100	70×100×100×100	7×25×25	175×25
	80.6%	83.7%	83.0%	82.1%	83.0%	83.3%

Table 4. Average pose accuracy obtained with different hierarchy settings. The models with 4-level label hierarchy are classification-only, *i.e*. the final regression layer is omitted.

Conditioning mechanism. By formulating the scene regression task as a coarse-to-fine joint classification-regression task can help break the complexity of the original regression problem to some extent, even without the proposed conditioning mechanism. To show this experimentally, we trained a variant of our network without the conditioning mechanism, *i.e*. we removed all the conditioning generators and layers, thus no coarse location information is fed to influence the network activations at the finer levels. We did preserve the coarse-to-fine joint learning, and still use the predicted location labels to determine the k-means cluster *w.r.t*. which the local regression coordinates are predicted. We denote this model variant by *Ours w/o cond*. In contrast to the regression-only baseline, the regression part still learns to perform local regression by predicting the offsets with respect to the cluster centers of the finest classification hierarchy. As shown in Fig. 3, this variant outperforms the regression-only baseline, and significant performance gain can be observed on the combined scenes. However, compared to our full architecture, it still falls far behind, especially on the largest i19-Scenes. This illustrates that the proposed conditioning mechanism plays a crucial role in our hierarchical coarse-to-fine scene coordinate learning framework, and the significantly improved performance compared to the regression-only baseline is not achievable without it.

Hierarchy and partition granularity. In Table 4 we report results obtained on the 7-Scenes and i7-Scenes datasets using label hierarchies of different depth and width. The results show that the performance of our approach is robust *w.r.t*. the choice of these hyperparameters, and only for the smallest 2-level label hierarchies that we tested we observed a significant drop in performance. Note that for the default setting (25×25), the results on 7-Scenes reported in Table 1 and 4 are the best across 10 runs of the randomly initialized k-means (mean $=$ 84.3%, SD $=$ 0.4%). How to optimally partition the scene could be explored in future work.

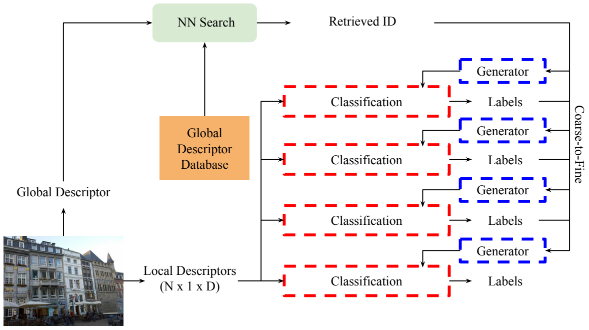

Figure 4. Illustration of our method with sparse local features and global image retrieval used in the Aachen dataset experiments.

4.5. Outdoor Aachen Localization Results

The Aachen dataset is a challenging outdoor large-scale dataset, which is particularly difficult for scene coordinate regression methods duo to the lack of dense model, the city-scale environment, and the night time queries. To the best of our knowledge, ESAC is the only existing method of this kind which gives reasonable results on this dataset.

We present a hybrid approach built upon our network for the challenging dataset. To resolve the sparsity of the training data, in [6], a re-projection error [5, 29] is optimized densely, which is not applicable to our method. Therefore, we resort to sparse local features [17, 19], such that during both training and test, our network only takes in a list of sparse features as input rather than a dense RGB image. To use image-level contextual information, we adopt an image retrieval technique. In addition to the location labels, every output layer including the first one is also conditioned on an image ID. During training, it is the ID of the training image. At test time, it is the ID of a retrieved image. We use SuperPoint [17] as the local feature, and NetVLAD [1] for global image retrieval. The results in Table 5 show that for the Aachen dataset the classification-only variant performs better, although it is not always the case, see Table 4. We use a 4-level classification-only network, and at the finest level, each cluster contains only one single 3D point. We use the retrieved database image also to perform a simple pre-RANSAC filtering step. Since the predictions are conditioned on the image ID, a prediction that is not visible in the corresponding image is likely to be a false match. Therefore, we filter out the predictions that are not visible in the corresponding retrieved image before the RANSAC stage. As shown in Table 5, this further improves the performance. Since the top-1 image can be a false positive, we run the pipeline for all the top-10 images, and select the prediction with the largest number of inliers. See the supplementary material for more details.

This approach significantly outperforms ESAC, and its performance is comparable to Active Search. However, compared to the hierarchical localization method of [41] which also uses SuperPoint and NetVLAD, our method still falls behind. Nevertheless, our method requires no database of local descriptors and the model size of our hierarchical network is 179MB, while in [41], a local descriptor database of 4GB is used. Our results reduce the gap between scene coordinate learning approaches and the state-of-the-art feature-based methods on this dataset, and we expect our method to perform better if a dense model is available. An advantage of the scene coordinate learning methods is that the model size does not grow linearly with the number of points in the scene model. This allows these methods to implicitly and efficiently store a dense descriptor point cloud in the network, and to produce dense matches at test time, which often leads to better pose estimation than sparse matches [54].

Method	Aachen Day 0.25m, 2° / 0.5m, 5° / 5m, 10°	Aachen Night 0.5m, 2° / 1m, 5° / 5m, 10°
AS [44]	57.3% / 83.7% / **96.6%**	19.4% / 30.6% / 43.9%
HL SP+NV [41]	**80.5%** / **87.4%** / 94.2%	**42.9%** / **62.2%** / **76.5%**
ESAC (50 experts) [6]	42.6% / 59.6% / 75.5%	3.1% / 9.2% / 11.2%
Ours top-10 w/ filt	71.1% / 81.9% / 91.7%	32.7% / 43.9% / 65.3%
Ours top-10 w/ filt w/o aug	65.5% / 77.3% / 88.8%	22.4% / 38.8% / 54.1%
Ours top-1 w/ filt	64.0% / 76.1% / 85.4%	18.4% / 32.7% / 53.1%
Ours top-1	58.3% / 66.4% / 80.2%	13.3% / 21.4% / 32.7%
Ours w/o retreived ID	50.6% / 56.3% / 70.1%	7.1% / 11.2% / 19.4%
Ours top-1 (4-level cls-reg)	47.8% / 61.8% / 79.9%	10.2% / 21.4% / 35.7%
Ours top-1 (3-level cls-reg)	20.9% / 42.2% / 76.9%	3.1% / 14.3% / 32.7%

Table 5. Accuracy on the Aachen dataset. Unless stated otherwise, we use a 4-level classification-only network for our method.

5. Conclusion

We have proposed a novel hierarchical coarse-to-fine scene coordinate learning approach, enabled by a FiLM-like conditioning mechanism, for visual localization. Our network has several levels of output layers with each of them conditioned on the outputs of the previous ones. Progressively finer localization labels are predicted with classification branches. The scene coordinate predictions can be obtained through a final regression layer or using the cluster centers at the finest level. The results show that the hierarchical scene coordinate network leads to more accurate camera re-localization performance than the previous regression-only approaches, achieving state-of-the-art results for single-image RGB localization on three benchmark datasets. Moreover, our novel architecture allows us to train compact models which scale robustly to large environments, achieving state-of-the-art on three combined scenes. Finally, we show a hybrid approach that further narrows the gap to the state-of-the-art feature-based methods for challenging large-scale outdoor localization.

Acknowledgements. This work has been supported by the Academy of Finland (grants 277685, 309902), and the French National Research Agency (grants ANR16-CE23-0006, ANR-11-LABX0025-01). We acknowledge the computational resources provided by the Aalto Science-IT project and CSC – IT Center for Science, Finland.

References

[1] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomás Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*, 2016. 2, 8, 13

[2] Vassileios Balntas, Shuda Li, and Victor Adrian Prisacariu. RelocNet: Continuous metric learning relocalisation using neural nets. In *ECCV*, 2018. 2

[3] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC - Differentiable RANSAC for camera localization. In *CVPR*, 2017. 1, 3

[4] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image. In *CVPR*, 2016. 3

[5] Eric Brachmann and Carsten Rother. Learning less is more - 6D camera localization via 3D surface regression. In *CVPR*, 2018. 1, 3, 5, 6, 7, 8, 11

[6] Eric Brachmann and Carsten Rother. Expert sample consensus applied to camera re-localization. In *ICCV*, 2019. 3, 5, 6, 8, 13

[7] Eric Brachmann and Carsten Rother. Neural-guided RANSAC: Learning where to sample model hypotheses. In *ICCV*, 2019. 3, 5, 6, 7

[8] Eric Brachmann and Carsten Rother. Visual camera re-localization from RGB and RGB-D images using DSAC. *arXiv:2002.12324*, 2020. 1, 3

[9] Samarth Brahmbhatt, Jinwei Gu, Kihwan Kim, James Hays, and Jan Kautz. Geometry-aware learning of maps for camera localization. In *CVPR*, 2018. 2

[10] Ignas Budvytis, Marvin Teichmann, Tomas Vojir, and Roberto Cipolla. Large scale joint semantic re-localisation and scene understanding via globally unique instance coordinate regression. In *BMVC*, 2019. 3

[11] Mai Bui, Shadi Albarqouni, Slobodan Ilic, and Nassir Navab. Scene coordinate and correspondence learning for image-based localization. In *BMVC*, 2018. 3

[12] Tommaso Cavallari, Luca Bertinetto, Jishnu Mukhoti, Philip Torr, and Stuart Golodetz. Let's take this online: Adapting scene coordinate regression network predictions for online RGB-D camera relocalisation. In *3DV*, 2019. 3, 5

[13] Tommaso Cavallari, Stuart Golodetz, Nicholas Lord, Julien Valentin, Victor Prisacariu, Luigi Di Stefano, and Philip HS Torr. Real-time RGB-D camera pose estimation in novel scenes using a relocalisation cascade. *PAMI*, 2019. 3, 5, 7

[14] Tommaso Cavallari, Stuart Golodetz, Nicholas A Lord, Julien Valentin, Luigi Di Stefano, and Philip HS Torr. On-the-fly adaptation of regression forests for online camera relocalisation. In *CVPR*, 2017. 3

[15] Wentao Cheng, Weisi Lin, Kan Chen, and Xinfeng Zhang. Cascaded parallel filtering for memory-efficient image-based localization. In *ICCV*, 2019. 2

[16] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). *arXiv:1511.07289*, 2015. 4, 5, 11

[17] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *CVPR Workshops*, 2018. 8, 13

[18] Mingyu Ding, Zhe Wang, Jiankai Sun, Jianping Shi, and Ping Luo. CamNet: Coarse-to-fine retrieval for camera relocalization. In *ICCV*, 2019. 2

[19] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A trainable CNN for joint detection and description of local features. In *CVPR*, 2019. 8

[20] Martin A Fischler and Robert C Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *CACM*, 24(6):381–395, 1981. 2

[21] Abner Guzmán-Rivera, Pushmeet Kohli, Ben Glocker, Jamie Shotton, Toby Sharp, Andrew W. Fitzgibbon, and Shahram Izadi. Multi-output learning for camera relocalization. In *CVPR*, 2014. 3

[22] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *ICRA*, 2016. 2

[23] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *CVPR*, 2017. 2

[24] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A convolutional network for real-time 6-DoF camera relocalization. In *ICCV*, 2015. 2, 5

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 11, 13

[26] 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

[27] Zakaria Laskar, Iaroslav Melekhov, Surya Kalia, and Juho Kannala. Camera relocalization by computing pairwise relative poses using convolutional neural network. In *ICCV Workshops*, 2017. 2

[28] Xiaotian Li, Juha Ylioinas, and Juho Kannala. Full-frame scene coordinate regression for image-based localization. In *RSS*, 2018. 2, 3, 7

[29] Xiaotian Li, Juha Ylioinas, Jakob Verbeek, and Juho Kannala. Scene coordinate regression with angle-based reprojection loss for camera relocalization. In *ECCV Workshops*, 2018. 3, 5, 8

[30] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 2

[31] Daniela Massiceti, Alexander Krull, Eric Brachmann, Carsten Rother, and Philip HS Torr. Random forests versus neural networks - What's best for camera localization? In *ICRA*, 2017. 3

[32] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Image-based localization using hourglass networks. In *ICCV Workshops*, 2017. 2

[33] Lili Meng, Jianhui Chen, Frederick Tung, James J Little, Julien Valentin, and Clarence W de Silva. Backtracking regression forests for accurate camera relocalization. In *IROS*, 2017. 3

[34] Lili Meng, Frederick Tung, James J Little, Julien Valentin, and Clarence W de Silva. Exploiting points and lines in regression forests for RGB-D camera relocalization. In *IROS*, 2018. 3

[35] Sven Middelberg, Torsten Sattler, Ole Untzelmann, and Leif Kobbelt. Scalable 6-DoF localization on mobile devices. In *ECCV*, 2014. 2

[36] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010. 5, 11

[37] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *AAAI*, 2018. 2, 3

[38] Noha Radwan, Abhinav Valada, and Wolfram Burgard. VLocNet++: Deep multitask learning for semantic visual localization and odometry. *RA-L*, 3(4):4407–4414, 2018. 2, 3, 5

[39] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. LCR-Net: Localization-classification-regression for human pose. In *CVPR*, 2017. 3

[40] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. LCR-Net++: Multi-person 2D and 3D pose detection in natural images. *PAMI*, 2019. 3

[41] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 2, 8, 13

[42] Paul-Edouard Sarlin, Frédéric Debraine, Marcin Dymczyk, Roland Siegwart, and Cesar Cadena. Leveraging deep visual descriptors for hierarchical efficient localization. In *CoRL*, 2018. 2

[43] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Fast image-based localization using direct 2D-to-3D matching. In *ICCV*, 2011. 1

[44] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *PAMI*, 39(9):1744–1756, 2016. 1, 2, 5, 8

[45] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6DoF outdoor visual localization in changing conditions. In *CVPR*, 2018. 2, 5

[46] Torsten Sattler, Akihiko Torii, Josef Sivic, Marc Pollefeys, Hajime Taira, Masatoshi Okutomi, and Tomas Pajdla. Are large-scale 3D models really necessary for accurate visual localization? In *CVPR*, 2017. 2

[47] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, 2012. 2, 5

[48] Torsten Sattler, Qunjie Zhou, Marc Pollefeys, and Laura Leal-Taixe. Understanding the limitations of CNN-based absolute camera pose regression. In *CVPR*, 2019. 2

[49] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 13

[50] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 1, 13

[51] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *CVPR*, 2013. 1, 2, 3, 4

[52] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014. 5, 11

[53] Linus Svärm, Olof Enqvist, Fredrik Kahl, and Magnus Oskarsson. City-scale localization for cameras with known vertical direction. *PAMI*, 39(7):1455–1461, 2016. 2

[54] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomás Pajdla, and Akihiko Torii. InLoc: Indoor visual localization with dense matching and view synthesis. In *CVPR*, 2018. 2, 5, 8

[55] Carl Toft, Erik Stenborg, Lars Hammarstrand, Lucas Brynte, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. Semantic match consistency for long-term visual localization. In *ECCV*, 2018. 2

[56] Abhinav Valada, Noha Radwan, and Wolfram Burgard. Deep auxiliary learning for visual localization and odometry. In *ICRA*, 2018. 2, 3, 5

[57] Julien Valentin, Angela Dai, Matthias Nießner, Pushmeet Kohli, Philip Torr, Shahram Izadi, and Cem Keskin. Learning to navigate the energy landscape. In *3DV*, 2016. 2, 4

[58] Julien Valentin, Matthias Nießner, Jamie Shotton, Andrew Fitzgibbon, Shahram Izadi, and Philip HS Torr. Exploiting uncertainty in regression forests for accurate camera relocalization. In *CVPR*, 2015. 3

[59] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using LSTMs for structured feature correlation. In *ICCV*, 2017. 2

[60] Philippe Weinzaepfel, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. Visual localization by learning objects-of-interest dense match regression. In *CVPR*, 2019. 3

[61] Luwei Yang, Ziqian Bai, Chengzhou Tang, Honghua Li, Yasutaka Furukawa, and Ping Tan. SANet: Scene agnostic network for camera localization. In *ICCV*, 2019. 2

—Supplementary Material—

In this supplementary material, we provide more details on network architecture, training procedure, and other experimental settings. Additional qualitative results are presented at the end.

A. Main Experiment Details

In this section, we present the experiment details on 7-Scenes, 12-Scenes, Cambridge Landmarks, and the combined scenes.

A.1. Network Architecture

We use a similar VGG-style [52] architecture as DSAC++ [5] as the base regression network, except we use ELU activation [16] instead of ReLU [36]. As mentioned in the main paper, we found that the plain regression network is faster to train with ReLU, while our network which has additional conditioning layers and classification branches works better with ELU. Conditioning layers and generators, as well as two additional classification branches, are added upon the base network for our 3-level hierarchical network which is used in the main experiments.

There are three convolutional layers with stride 2 in the regression base network. The output resolution of the regression branch is thus reduced by a factor of 8. Strided convolution, dilated convolution and upconvolution are used in the two classification branches to enlarge the receptive field and preserve the output resolution. The predicted classification labels are converted into one-hot format before being fed into the generators. If more than one label map used as input to a conditioning generator, the label maps are concatenated.

The detailed architecture is given in Fig. 5. For experiments on 7-Scenes, 12-Scenes, Cambridge Landmarks, we use the same network architecture. For experiments on the combined scenes, we increased the number of channels for certain layers and added two more layers in the first conditioning generator. The additional layers are marked in red, and the increased channel counts are marked in red, blue and purple for i7-Scenes, i12-Scenes and i19-Scenes, respectively. The more compact architecture for the experiments (*Ours capacity-* in Table 2 and Fig. 3 of the main paper) on the combined scenes is illustrated in Fig. 6. In the case we use different numbers of channels for a convolutional layer, the channel counts are marked in red, blue and purple for i7-Scenes, i12-Scenes and i19-Scenes respectively.

As in DSAC++ [5], our network always takes an input image of size 640 × 480. We follow the same practice to resize larger images as [5]. That is, the image is first rescaled to height 480. If its width is still larger than 640, it is cropped to width 640. Central cropping is used at test time, and random horizontal offsets are applied during training.

A.2. Network Training

For 7-Scenes and 12-Scenes, our network is trained from scratch for 300K iterations with an initial learning rate of 10^{-4} using Adam [25], and the batch size is set to 1. We halve the learning rate every 50K iterations for the last 200K iterations. For the Cambridge Landmark dataset, the dense reconstructions are far from perfect. The rendered ground truth scene coordinates contain a significant amount of outliers, which make the training difficult. Therefore, we train the network for 600K iterations for experiments on this dataset. For the combined scenes, the network is trained for 900K iterations.

As mentioned in the main paper, we found that the accuracy of the final regression predictions is critical to high localization performance. Therefore, a larger weight is given to the regression loss term. The weights for the classification loss terms w_1, w_2 are set to 1 for all scenes. The weight for the regression loss term is set to 100,000 for the three combined scenes and 10 for the other datasets.

For data augmentation, affine transformations are applied to each training image. We translate, rotate, scale, shear the image by values uniformly sampled from [-20%, 20%], [$-30°$, $30°$], [0.7,1.5], [$-10°$, $10°$], respectively. In addition, we also augment the images with additive brightness changes uniformly sampled from [-20, 20]. When training without data augmentation, as with [5], we randomly shift the image by -4 to 4 pixels, both horizontally and vertically, to make full use of data, as the output resolution is reduced by a factor of 8.

A.3. Pose Optimization

At test time, we follow the same PnP-RANSAC pipeline and parameter settings as in [5]. The inlier threshold is set to $\tau = 10$ for all the scenes. The softness factor is set to $\beta = 0.5$ for the soft inlier count [5]. A set of 256 initial hypotheses are sampled, and the refinement of the selected hypothesis is performed until convergence for a maximum of 100 iterations.

A.4. Run Time

The network training takes ≈12 hours for 300K iterations on an NVIDIA Tesla V100 GPU, and ≈18 hours on an NVIDIA GeForce GTX 1080 Ti GPU.

At test time, it takes ≈100ms for our method to localize an image on an NVIDIA GeForce GTX 1080 Ti GPU and an Intel Core i7-7820X CPU. Scene coordinate prediction takes 50-65ms depending on the network size. Pose optimization takes 30-60ms depending on the accuracy of the predicted correspondences.

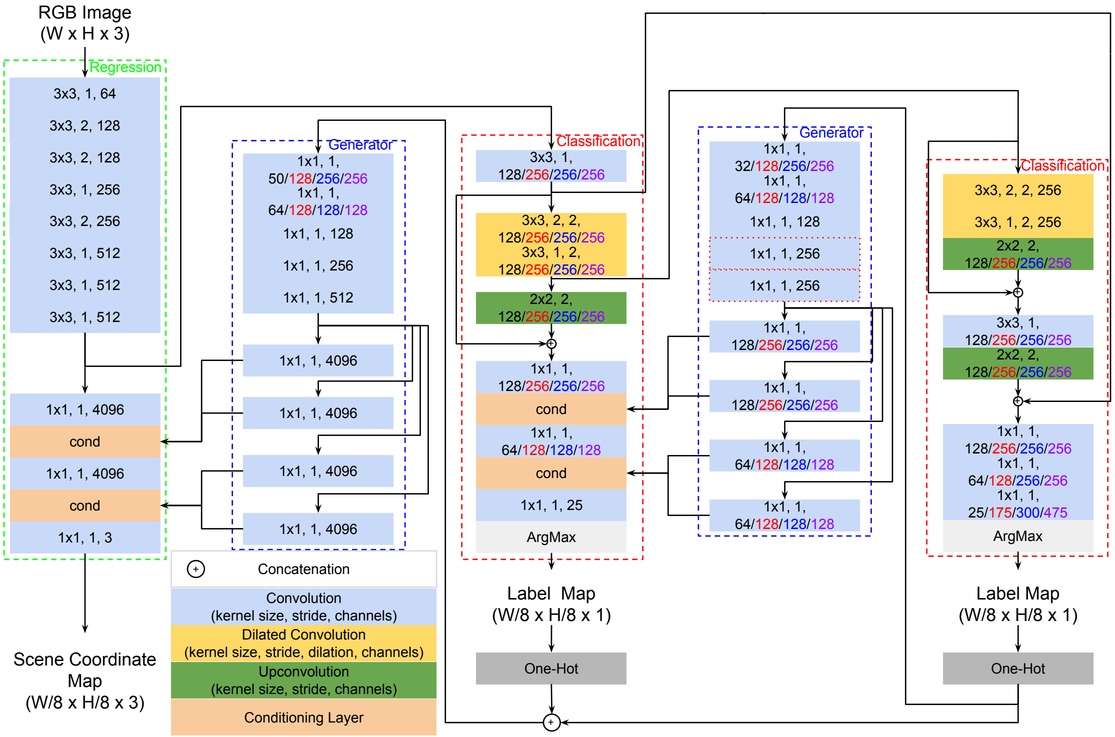

Figure 5. Detailed network architecture.

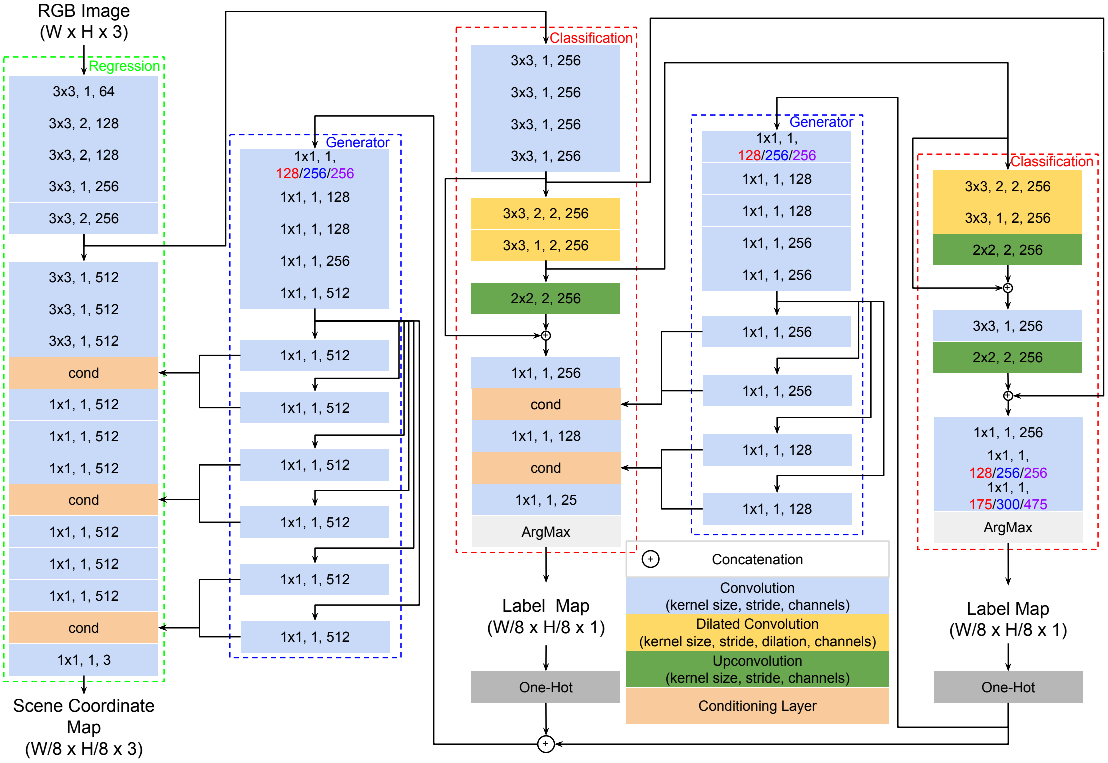

Figure 6. The more compact architecture of *Ours capacity-*.

B. Experiments on the Aachen Dataset

In this section, we provide the experimental details on the Aachen dataset.

B.1. Ground Truth Labels

Similar to the experiments on the other datasets, to generate the ground truth location labels, we run hierarchical k-means clustering on the sparse point cloud model used in [41], which is built with COLMAP [49, 50] using SuperPoint [17] as local feature detector and descriptor. For this dataset we adopt a 4-level classification-only network. We also experimented with two classification-regression networks, but the 4-level classification-only network works better (see Table 5 in the main paper). For the 4-level classification-only network, we use four-level hierarchical k-means with the branching factor set to 100 for all levels. This results in ≈685K valid clusters at the finest level, with each of them containing only a single 3D point. For the experiments with the 4-level classification-regression network and the 3-level classification-regression network, we use three-level and two-level hierarchical k-means with the same branching factor setting (100 for all levels), respectively.

B.2. Network Architecture

As stated in the main paper, for the experiments on the Aachen dataset, we use a list of sparse features as input to the network, rather than a regular RGB image. Due to the sparse and irregular format of the input, we use 1×1 convolutional layers in the network. We add a dummy spatial dimension to the input, *i.e.* we use a descriptor map of size N×1×256 as input. In addition, there are no shared layers between different levels. To use image-level contextual information, every output layer including the first one is also conditioned on an image ID. To achieve this, the encoded image ID is concatenated with the label maps (if available) and then fed into the conditioning parameter generators. As mentioned in the main paper, during training, we use the ID of the training image. At inference time, we adopt NetVLAD [1] for global image retrieval, and we use the ID of a retrieved image. The detailed architecture of the 4-level classification-only network is given in Fig. 7. For the 4-level classification-regression network, we simply change the last classification layer to a regression output layer. For the 3-level classification-regression network, one classification level is further removed.

B.3. Network Training

The network is trained from scratch for 900K iterations with an initial learning rate of 10^{-4} using Adam [25], and the batch size is set to 1, similar to the previous experiments. We halve the learning rate every 50K iterations for the last 200K iterations. As in [6, 41], all images are converted to grayscale before extracting the descriptors. Random affine transformations, brightness and contrast changes are also applied to the images before the feature extraction. During training, we ignore the interest point detection, and a descriptor is extract from the dense descriptor map if it has an available corresponding 3D point in the spare 3D model. Following [41], before extracting the NetVLAD [1] and SuperPoint [17] features, the images are downsampled such that largest dimension is 960. At test time, for SuperPoint, Non-Maximum Suppression (NMS) with radius 4 is applied to the detected keypoints and 2K of them with the highest keypoint scores are used as the input to our network.

B.4. Pose Optimization

We follow the PnP-RANSAC algorithm as in [41] and the same parameter settings are used. The inlier threshold is set to $\tau = 10$, and at most 5,000 hypotheses are sampled if no hypotheses with more than 100 inliers are found. Note that the pose optimization is applied independently for all the top-10 retrieved database images.

B.5. Run Time

The network training takes 2-3 days on an NVIDIA Tesla V100/NVIDIA GeForce GTX 1080 Ti GPU. On an NVIDIA GeForce GTX 1080 Ti GPU and an Intel Core i7-7820X CPU, it takes ≈1.1/1.4s (Aachen Day/Aachen Night) for our method to localize an image. It takes ≈170ms to extract the global and local descriptors. It takes ≈280ms (10×28ms) for scene coordinate prediction and ≈600/900ms (10×60/90ms) (Aachen Day/Aachen Night) for pose optimization. The time needed for global descriptor matching and the simple pre-RANSAC filtering is negligible.

C. Additional Qualitative Results

We show in Fig. 8 the quality of scene coordinate predictions for test images from 7-Scenes/i7-Scenes, and compare our method to the regression-only baseline. The scene coordinates are mapped to RGB values for visualization.

We show in Fig. 9 the quality of scene coordinate predictions for the Aachen dataset experiments. The scene coordinate predictions are visualized as 2D-2D matches between the query and database images. We show only the inlier matches.

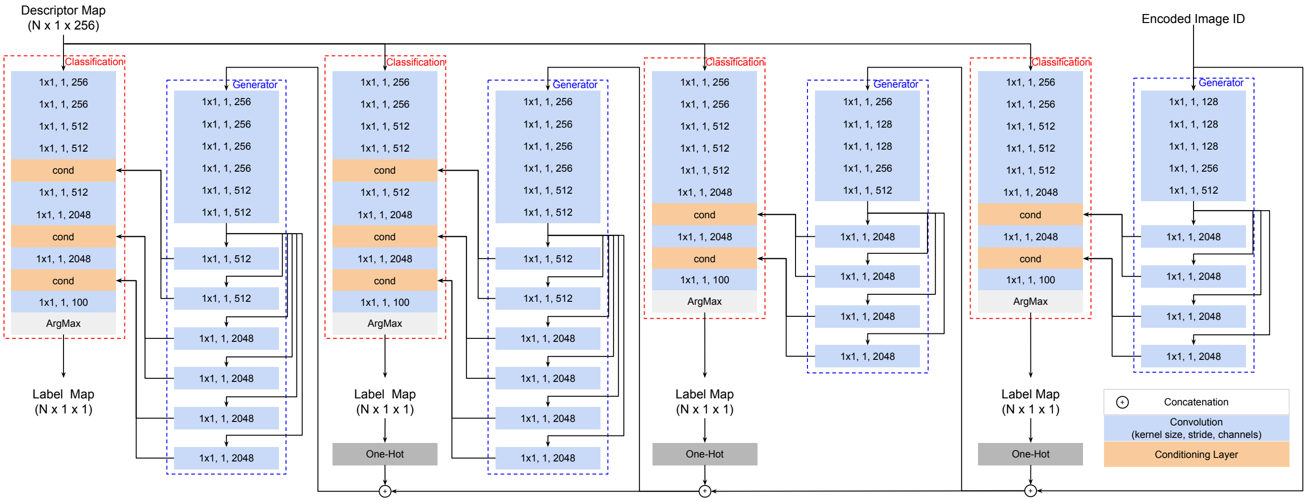

Figure 7. Detailed network architecture of the 4-level classification-only network for the Aachen dataset experiments.

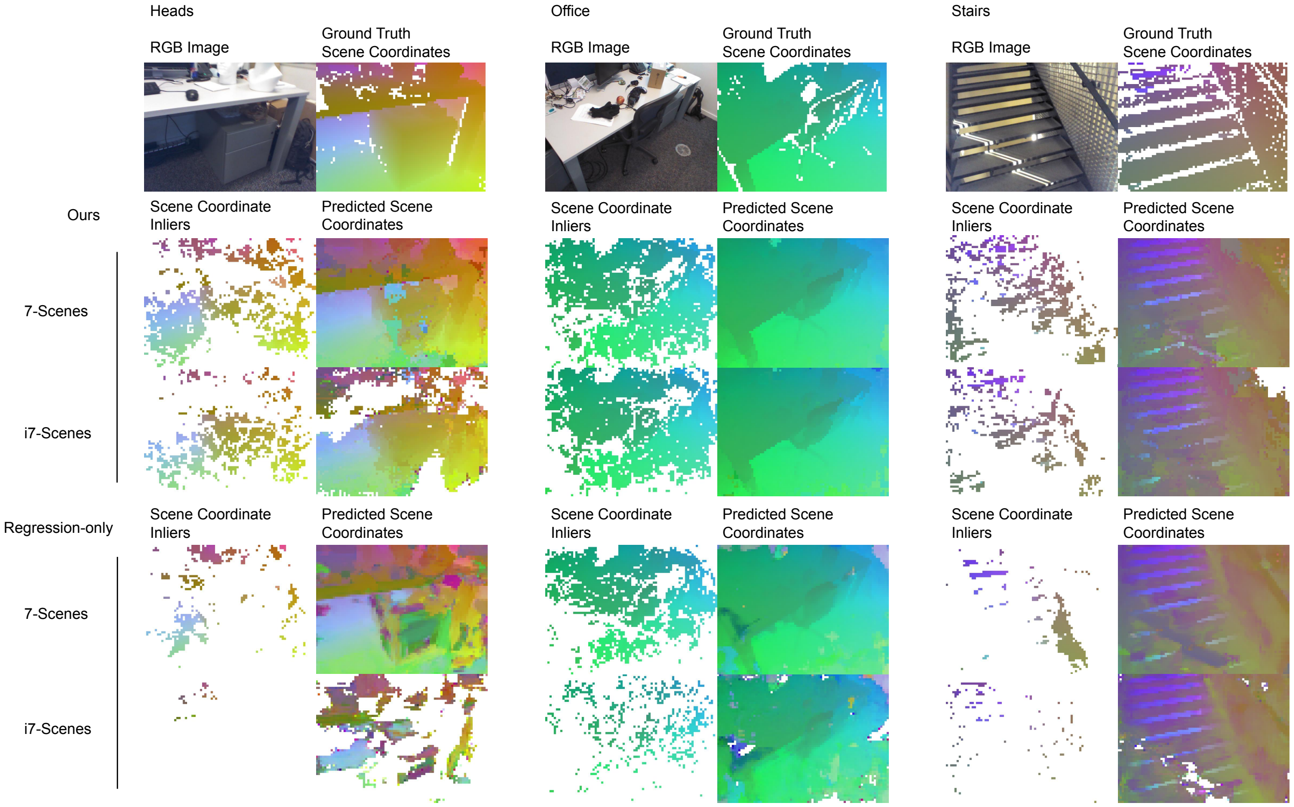

Figure 8. We visualize the scene coordinate predictions for three test images from 7-Scenes/i7-Scenes. The XYZ coordinates are mapped to RGB values. The ground truth scene coordinates are computed from the depth maps, and invalid depth values (0, 65535) are ignored. Should a scene coordinate prediction be out of the scope of the corresponding individual scene, the prediction is treated as invalid and not visualized. We also visualize the scene coordinate inliers retained after the pose optimization (PnP-RANSAC) stage. On both 7-Scenes and i7-Scenes, our method produces consistently better scene coordinate predictions with more inliers compared to the regression-only baseline.

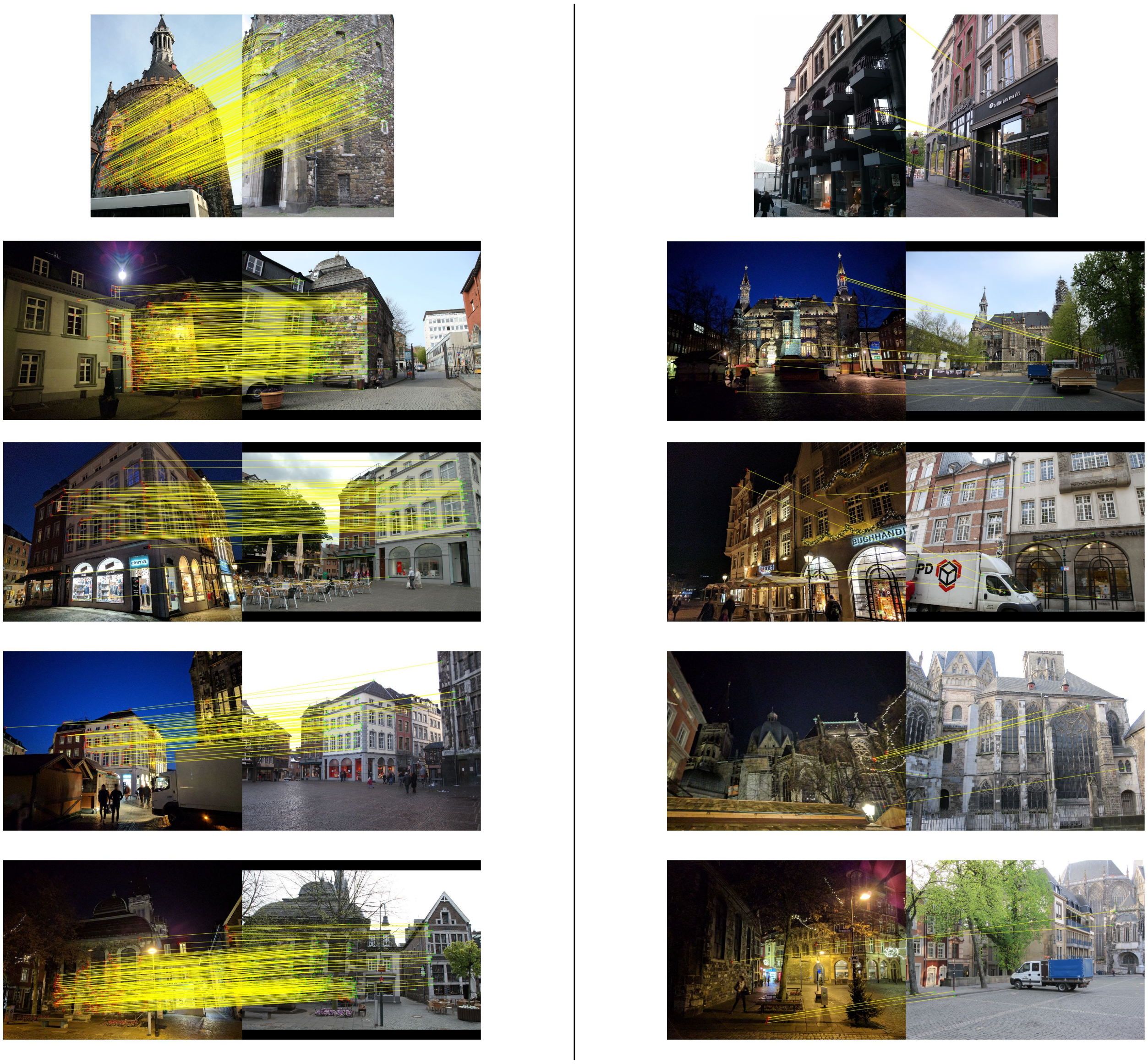

Figure 9. We show the scene coordinate predictions for the Aachen dataset experiments. The scene coordinate predictions are visualized as 2D-2D matches between the query (left) and database (right) images. For each pair, the retrieved database image with the largest number of inliers is selected, and only the inlier matches are visualized. We show that our method is able to produce accurate correspondences for challenging queries (left column). Failure cases are also given (right column).