-Geometric Image Correspondence Verification by Dense Pixel Matching-

Appendix

In this appendices we show additional qualitative and quantitative results of the proposed approach. In Sec. B we provide an ablation study and analyze the influence of different design choices of our method to the localization performance. We demonstrate the benefits of the unified correspondence map decoder (UCMD) compared to the architecture with multiple decoders in Sec. C Finally, qualitative localization and pixel correspondence estimation results are shown in Sec. E.

A. Additional Baselines

In this work, we propose two similarity functions for geometric verification, *i.e.*:

$$S = \frac{C}{I} \cdot exp\left(-\frac{W \cdot H}{C}\right), \tag{1}$$

$$S_F = \overbrace{\log_{10}\left(S_L \cdot S\right)}^{R} \cdot \underbrace{10^{-G}}_{Q} \tag{2}$$

where I and C is the number of inliers and cyclically consistent inliers between two (W, H) images (A, B), respectively; $S_L = \sum_a \left(f_A^a \cdot f_B^a\right) m^a$ is the local similarity between each hypercolumn (f_A^a and f_B^a) of the NetVLAD [1] image descriptor at location a; G is the global similarity value.

We compare our method with two baselines: *i)* recently proposed geometric verification pipeline Inloc [10], and *ii)* a neural network based method that learns the scoring functions, S and S_F given $\{C, I\}$, and $\{C, I, S_L, G\}$ as input. We present more details about the baselines next.

Inloc. Inloc is a indoor localization pipeline consisting of three primary stages: **i**) ranking of database images by measuring global representation similarity with a given query. The global representations are obtained from the image retrieval pipeline, *e.g.* NetVLAD [1]; **ii**) a shortlist of top ranked database images are re-ranked based on geometric verification using dense CNN descriptors. The dense descriptors are obtained from different layers of the NetVLAD pipeline followed by a coarse to fine matching using nearest-neighbor search. The geometric verification is done using a standard RANSAC based inlier count. The final score is the sum of global similarity and inlier count; **iii**) the top ranked geometrically verified database images are fed into a pose verification stage. The final stage first estimates candidate query poses w.r.t. the current shortlisted database images. The estimated pose is then verified using view synthesis, a process requiring dense database depth maps. Our proposed geometric verification pipeline is similar to Inloc components i) and ii). The pose verification stage requires depth maps which is not always available. Therefore, we evaluate Inloc pipeline until the geometric verification stage and report results in Tab. 1b.

Learnt similarity functions. Since both Eq. 1 and Eq. 2 are hand-crafted we provide a FCNN-based model that can *learn* the similarity function. More specifically, we experiment with two independent models (for S and S_F) which can predict whether two images similar or not based on C, I, S_L, and G. Both models have similar architectures $FC(N, 128) - ReLU - FC(128, 128) - ReLU - FC(128, 1)$, where the shorthand notation is used was the following: FC is a fully connected linear layer; N is the number of input units (2 $\{C, I\}$ for S and 4 $\{C, I, S_L, G\}$ for S_F, respectively). We refer to these models as S-FCNN and S_F-FCNN. Both models have been trained by minimizing binary cross-entropy loss function in a supervised manner.

Results. We now compare S and S_F with Inloc geometric verification pipeline on Tokyo247 dataset. Results demonstrate that our proposed function S and S_F outperform Inloc across all Recall rates as shown in Tab. 1a. We observed that for many query-database image pairs, Inloc fails to find any inliers. This can be attributed to significant clutter, illumination change (day-night) and occlusion in this challenging dataset. The learnt similarity functions S-FCNN and S_F-FCNN have very promising results and perform better than NetVLAD. In particular, S-FCNN has comparable performance to the proposed S. However, S_F-FCNN could not achieve any improvement compared to S-FCNN. We leave further analysis for future work.

B. Ablation study

In this section we perform an ablation study on the proposed equations Eq. 1 and Eq. 2 for geometric verification.

Methods	Recall		
	r@1	r@5	r@10
Inloc [10]	62.54	67.62	70.48
NetVLAD-Pitts [1]	61.27	73.02	78.73
DenseVLAD [11]	67.10	74.20	76.10
S-FCNN	67.94	81.90	85.08
S_F-FCNN	63.49	81.59	85.71
Proposed (S) Pitts	71.43	82.54	85.08
Proposed (S_F) Pitts	**77.14**	**84.44**	**86.67**

(a) The proposed similarity functions S and S_F perform better strong baseline methods.

Methods	Recall		
	r@1	r@5	r@10
NetVLAD-Pitts [1]	61.27	73.02	78.73
I (inliers)	56.83	78.41	83.81
C (cyclically consistent inliers)	70.16	82.86	85.71
C/I	64.76	82.54	85.71
Proposed (S) Pitts	71.43	82.54	85.08

(b) Localization performance on the Tokyo247 dataset (higher is better).

Methods	Recall		
	r@1	r@5	r@10
NetVLAD-Pitts [1]	61.27	73.02	78.73
$\log_{10}(S_L * S)$	73.65	83.49	86.67
G	69.84	80.95	85.08
Proposed (S_F) Pitts	**77.14**	**84.44**	**86.67**

(c) Localization performance on the Tokyo247 dataset (higher is better).

R \ Q	$5/G$	10^{-G}
S_L	77.78	81.34
$S_L * S$	82.04	83.70
$\log_{10}(S_L * S)$	85.37	**85.94**

(d) Localization performance (Recall@1) on the *Pittsburgh* test dataset (higher is better). We analyze the performance of different Q and R of the original similarity function (2). The baseline, NetVLAD achieves 81.59 % Recall@1

Table 1: **Ablation study.** We evaluate the proposed similarity functions S and S_F with different settings on Tokyo24/7 and Pittsburgh datasets.

For Eq. 1, we analyze the impact of each variable, C, I on retrieval performance on Tokyo247 dataset independently. The results are presented in Tab. 1b.

Results. First we provide the ablation study for Eq. 1. Results demonstrate that simple Inlier count performs worse than the baseline NetVLAD and our proposed S at Recall@1. However, the retrieval performance improves over NetVLAD for Recall@5 and Recall@10. Cyclically consistent inliers C outperform NetVLAD across various Recall rates. Similarly, the ratio C/I performs marginally better but it falls slightly behind of C for Recall@1 (by about 6 %). Both C and C/I perform on par with the proposed S across Recall@5 and Recall@10. However, S has a clear performance advantage over C and C/I for Recall@1 as shown in Tab. 1b.

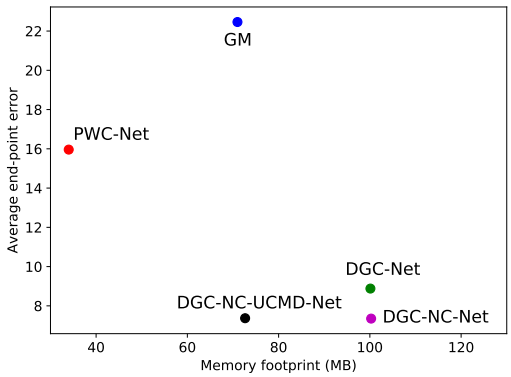


Figure 1: **AEPE averaged over all HPatches [2] sequences versus memory footprint**. Accuracy of both proposed methods (DGC-NC-Net and DGC-NC-UCMD-Net) is about on par, however, UCMD allows to decrease memory footprint by 30%.

Now, we perform an ablation study for Eq. 2. As mentioned in the main manuscript, the proposed S_F is used to re-rank the top 20 database images in the shortlist, L as ranked by S. Here, we perform the final re-ranking using just the local descriptor similarity component, $\log_{10}(S_L * S)$, and global representation distance, G. Results in Tab. 1c demonstrate that re-ranking with G decreases retrieval performance compared to the initial ranking by S. On the other hand, local descriptor similarity S_L weighted by S significantly improves over the baselines and initial ranking by S. However, the proposed combination of local and global representation similarity outperforms each individual component across all Recall rates.

The key idea here is to combine the similarity functions, S_L, S and G. It is important to note that S_L and S are similarity functions, while G is a distance function, hence, it is inversely proportional to global similarity. The inversely proportional functions, $R(S_L, S)$ and $Q(G)$ can be combined in many different ways. We present a few in Tab. 1d. The co-efficient (5 and 10) associated with G in the columns of the Tab. 1d have been obtained using a grid search over the range $(1, 10000)$ on Pittsburgh test dataset. In addition, we found $\hat{S}_F = R * Q$ performs clearly better than $\hat{S}_F = R + Q$. Hence, we only present results for various R and Q for $\hat{S}_F = R * Q$ in Tab. 1d. The precise form of the combination of these similarity functions has been obtained based on validation experiments on test set of Pittsburgh dataset. Tab. 1d shows that various possible combinations give better performance than NetVLAD which achieves 81.59 at Recall@1.

C. The benefits of UCMD

As shown in the main manuscript, we propose the unified correspondence map decoder which leads to a compact but efficient architecture. In addition to ablation study presented in the main part, here we report the average end point error averaged over all sequences of the HPatches [2] dataset obtained by the proposed approach and each strong baseline method (PWC-Net [9], geometric matching GM [7], and DGC-Net [5]) and allocated GPU memory. The results are illustrated in Fig. 1. In contrast to DGC-NC-Net with 5 separate decoders, the proposed UCMD can significantly decrease memory footprint (by 30%) achieving comparable accuracy.

The amount of memory allocated by GM [7], DGC-Net [5], DGC-NC-Net, and DGC-NC-UCMD-Net is higher compared to PWC-Net since all those models have used pre-trained VGG-16 network as encoder. Therefore, in addition to memory consumption and total number of parameters given in the main manuscipt, we compute the total number of *learnable* parameters of each model and provide the results in Tab. 2.

Model	Number of learnable parameters
PWC-Net [9]	8 749 280
GM [7]	3 271 576
DGC-Net [5]	2 675 338
Proposed (DGC-NC-Net)	2 685 079
Proposed (DGC-NC-UCMD-Net)	**940 561**

Table 2: Number of learnable parameters of two proposed architectures and strong baseline methods.

D. Implementation details

We train our network end-to-end using Adam [4] solver with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. As a preprocessing step, the training images are resized to 240×240 and further mean-centered and normalized using mean and standard deviation of ImageNet dataset [3]. We use a batch size of 32, an initial learning rate of 10^{-2} which is gradually decreased during training. The weight decay is initialized to 10^{-5} in all experiments and no dropout was used in our experiments. Our method is implemented using PyTorch framework [6] and trained on two NVIDIA Titan X GPUs.

Localization. For image retrieval and localization experiments, we resized the input images to 640×480 and extracted descriptors (feature maps) from the output of the *conv3* and *conv5* layer in the NetVLAD architecture[1]. The low resolution feature maps were then respectively upsampled using bilinear interpolation to 160×120 and concatenated along the channel dimension. We then compute local descriptor similarity, S_L (Eq. 2 in main paper). As this stage requires dense correspondence map, we resized the corresponding output of DGC-NC-UCMD-Net to 160×120. Note that in Eq. 2 (in main paper), we first extract the descriptors from original images, and then warp the feature maps to compute similarity.

Computational time. We evaluate the computation time on Tokyo24/7 dataset for the proposed method. It takes on average 0.4s to re-rank a database image for a given query using Eq. 1. One pass through the network requires 0.08s, while RANSAC on the output dense map involves major computations requiring 0.32s. Further re-ranking (Eq. 2) takes 0.04s as it uses the results (dense correspondence maps) from Eq. 1.

E. Qualitative results

Localization (image retrieval) performance. Fig. 2 reports an additional set of results obtained for the Tokyo24/7 dataset. Namely, it includes top-1 Nearest Neighbour (Recall@1 metric) obtained by NetVLAD [1] and our approach, respectively, for a given query. It clearly shows the proposed method improves retrieval results compared to NetVLAD and can cope with major changes in appearance (illumination changes in the scene) between the database and query images. Qualitative image retrieval results on Aachen Day-Night [8] are illustrated in Fig. 3a.

Dense pixel correspondences are presented in Fig. 3. Each row shows one test pair from the Aachen Day-Night and Tokyo24/7 datasets, respectively. Ground truth matching keypoints are illustrated in different colors and have been used *only* for pixel correspondence evaluation. Keypoints of the same color are supposed to match each other. We manually indicated 3 keypoints in the target image for visualization purposes and the corresponding locations in the source image have been obtained by the proposed automatic dense matching approach. That is, given an input image pair (source and target images), our method predicts the correspondence map which is then used to obtain the location of keypoints. The results demonstrate that the proposed method can handle such challenging cases as different illumination (day/night) conditions, occlusions, and significant viewpoint changes producing accurate pixel correspondences.

F. Limitations and future directions

We have demonstrated that the proposed method can localize queries under challenging conditions but it fails for very large viewpoint change (*e.g.* 180^o rotation while observing the same place) and significant scale change. In addition, it would be interesting to propose an end-to-end semi-supervised approach which can efficiently learn similarity functions.

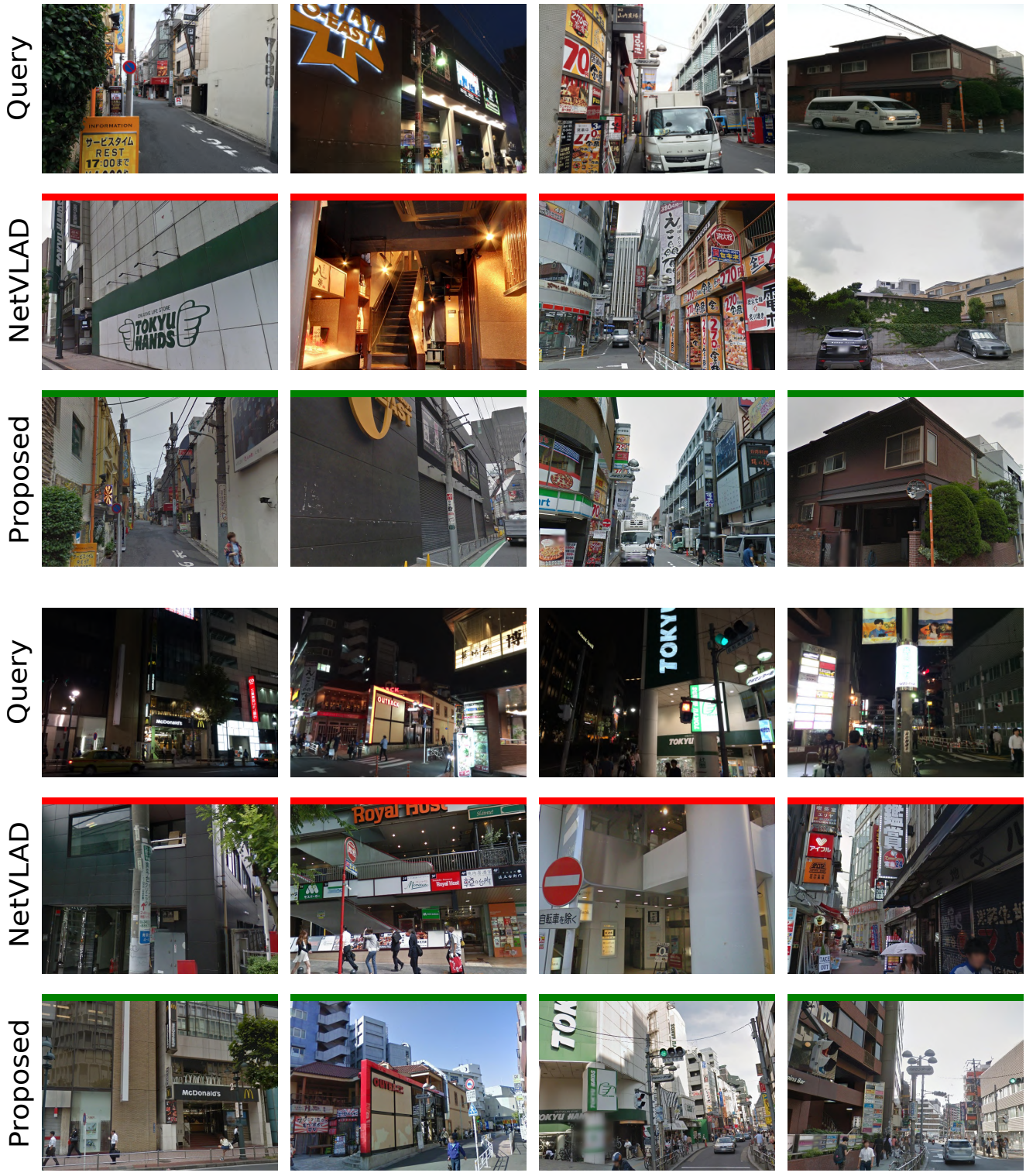
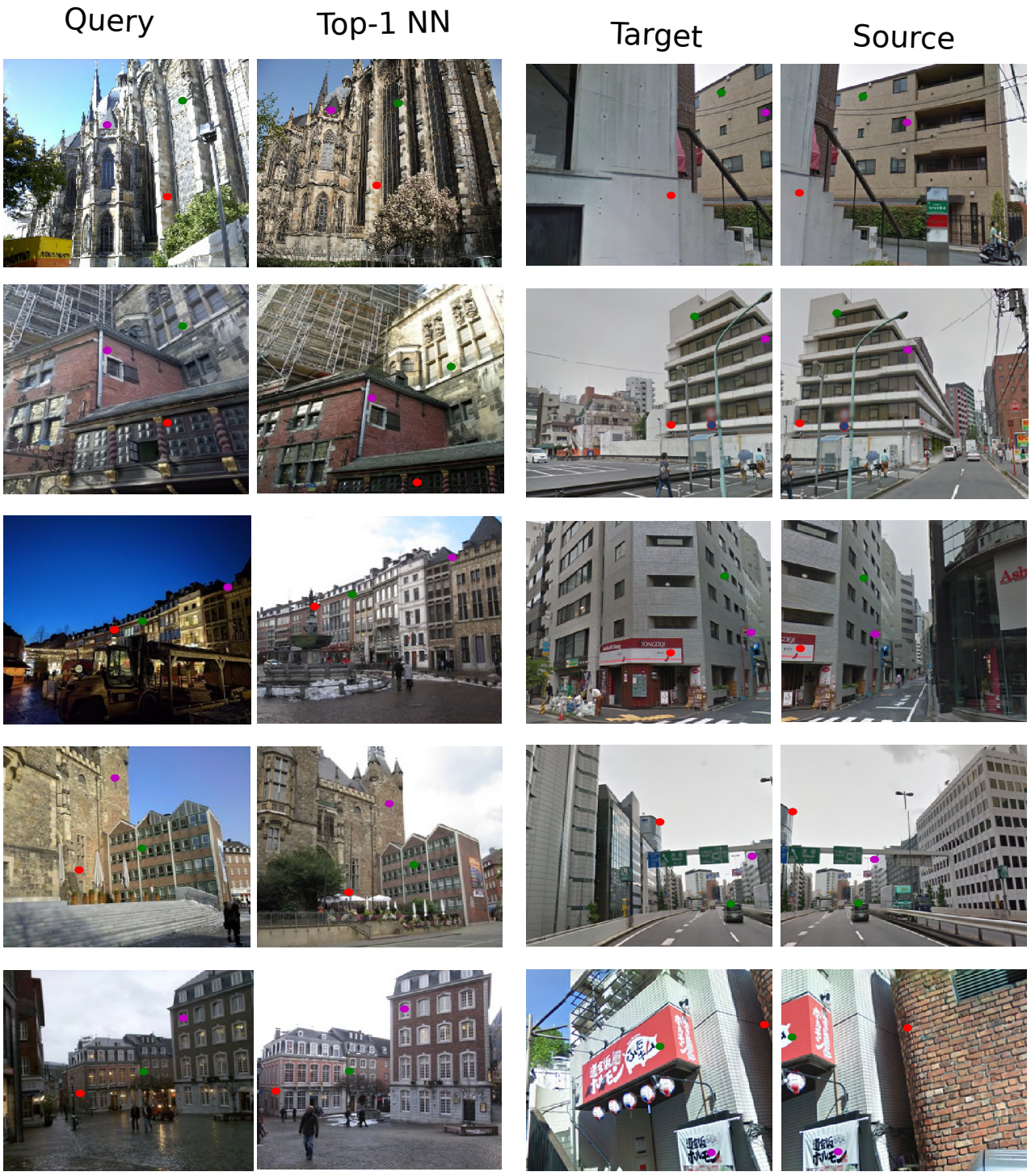


Figure 2: **Qualitative results** produced by NetVLAD [1] (rows 2 and 5) and the proposed method (rows 3 and 6) on Tokyo24/7 [11]. Each column corresponds to one test case: for each query (row 1 and 4) top-1 (Recall@1) nearest database image has been retrieved. The green and red strokes correspond to correct and incorrect retrieved images, respectively. The proposed approach can handle different illumination conditions (day/night) and significant viewpoint changes.



(a) Retrieval performance and pixel correspondences on Aachen Day-Night

(b) Pixel correspondences on Tokyo24/7

Figure 3: **Qualitative image retrieval 3a and dense pixel correspondence estimation results** produced by the proposed approach. We evaluate our approach on two challenging datasets: Tokyo24/7 and Aachen Day-Night. More image retrieval results are illustrated in Fig. 2. Each row of Fig. 3b corresponds to one test case. Ground truth keypoints have been manually selected in the target image for visualization purposes and the corresponding locations in the source image are obtained by the proposed dense matching method. Keypoints of the same color are supposed to match each other.

References

[1] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proc. CVPR*, 2016. 1, 2, 3, 4

[2] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proc. CVPR*, 2017. 2, 3

[3] J. Deng, W.Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009. 3

[4] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. ICLR*, 2014. 3

[5] I. Melekhov, A. Tiulpin, T. Sattler, M. Pollefeys, E. Rahtu, and J. Kannala. DGC-Net: Dense Geometric Correspondence Network. In *Proc. WACV*, 2019. 3

[6] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. 2017. 3

[7] I. Rocco, R. Arandjelović, and J. Sivic. Convolutional neural network architecture for geometric matching. In *Proc. CVPR*, 2017. 3

[8] T. Sattler, W. Maddern, A. Torii, J. Sivic, T. Pajdla, M. Pollefeys, and M. Okutomi. Benchmarking 6DOF Urban Visual Localization in Changing Conditions. In *Proc. CVPR*, 2018. 3

[9] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proc. CVPR*, 2018. 3

[10] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. In *Proc. CVPR*, 2018. 1, 2

[11] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 Place Recognition by View Synthesis. In *Proc. CVPR*, 2015. 2, 4