A. Supplementary appendix

A.1 Implementation details

Network details. In terms of the matchability predictor, we construct 4-layer MLPs whose output node numbers are 128, 32, 32, 1, respectively. The visual context encoder is composed of two 2-layer MLPs, located before/after the concatenation with raw local features. We insert context normalization only into the former MLPs, while insertion in the latter one is observed to harm the performance.

Performance of the retrieval model. The retrieval model is trained on *Google-Landmarks Dataset* [28], which contains more than 1M landmark images. In our experiments, we have compared different networks for the retrieval performance. In brief, ResNet-101 is slightly better than ResNet-50, while VGG and AlexNet are notably underperforming. We choose ResNet-50 for better tradeoffs in memory and speed.

Instead of adopting the training scheme in [33], we find that the model pretrained on landmark classification task (containing 15K classes) as in [28] suffices to produce satisfactory results in practice, and avoids difficulties on preparing training data for Siamese networks or hard negative mining with complex heuristics. We have evaluated the retrieval model with MAC aggregation on standard Oxford buildings dataset [29], where we obtain mAP of 0.83, on par with [33] of 0.80.

Keypoint coordinate augmentation. Similar to [6], we choose to use 4-point parameterization to represent the homography as follows:

$$H_{4point} = \begin{Bmatrix} u_1 + \triangle u_1 & v_1 + \triangle v_1 \\ u_2 + \triangle u_2 & v_2 + \triangle v_2 \\ u_3 + \triangle u_3 & v_3 + \triangle v_3 \\ u_4 + \triangle u_4 & v_4 + \triangle v_4 \end{Bmatrix},$$

where $(u_1, v_1), (u_2, v_2), (u_3, v_3), (u_4, v_4)$ are four corner points at $(-1, 1), (1, 1), (-1, -1), (1, -1)$, and $\triangle u_i, \triangle v_i$ are random variables between $(-0.5, 0.5)$. One can easily convert H_{4point} to a standard 3×3 homography by, e.g., normalized Direct Linear Transform (DLT) algorithm. In our implementation, we apply the random homography on each keypoint coordinate set before feeding it into the geometric context encoder.

Learning with noisy keypoints. The training of proposed framework, apparently, needs to be conducted between image pairs instead of isolated patches, since we also take keypoint coordinates as input. Such data organization is referred to as simulating image matching in [23]. However, the simulation in [23] is not complete, as it considers only keypoints that have successfully established correspondences, whereas in real situation, only a subset of keypoints is repeatable in other images. In practice, as illustrated in Fig. 6, we divide keypoints obtained from SfM as in [48, 23] into three categories: i) *Matchable*: repeatable and verified by SfM; ii) *Undiscovered*: repeatable but did not survive the SfM. iii) *Unrepeatable*: unable to be redetected in other images.

In the training, we randomly sample a number of matchable keypoints as well as some undiscovered and unrepeatable keypoints (denoted as *noisy keypoints*), in order to have a complete simulation that is necessary to acquire strong generalization ability. Otherwise, the training will consider only ideal setting with all matchable keypoints, which is inconsistent with real applications.

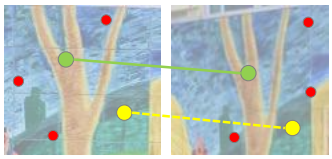

Figure 6: We divide keypoints after SfM into three categories: matchable (green), undiscovered (yellow) and unrepeatable (red), and aim to perform a complete simulation in training that incorporates all three types of keypoints.

To incorporate with the above training strategy, we need to make some adaptation on the loss formulation. Formally, given index sets $C_m = \{i_1, ..., i_{K_m}\}$ and $C_n = \{i_1, ..., i_{K_n}\}$, where K_m and K_n are numbers of matchable and noisy keypoints for an image pair, the losses of Eq. 3 and Eq. 5 are now rewritten as:

$$\mathcal{L}'_{quad} = \frac{1}{K_m(K_m - 1)} \sum_{i,j \in C_m, i \neq j} \max(0, 1 - R(\boldsymbol{f}_1^i, \boldsymbol{f}_1^j, \boldsymbol{f}_2^i, \boldsymbol{f}_2^j)), \tag{8}$$

and

$$\mathcal{L}'_{N\text{-}pair} = -\frac{1}{2}\Big(\sum_{i \in C_m} \log s_{ii}^r + \sum_{i \in C_m} \log s_{ii}^c\Big). \tag{9}$$

Subsequently, adding noisy keypoints will first influence the encoding of geometric context, posing a harder training settings and playing a key role in order to acquire the invariance properties. Second, it will influence the computation of $\mathcal{L}_{N\text{-}pair}$, as those keypoints will be all cross-paired as negative samples. It also enables us to increase the pair number in each batch, i.e., 1024 in our implementation compared with 64 in GeoDesc [23], which boost the effectiveness of N-pair loss as observed in [25].

Further joint processing in aggregation step. In this work, as in Sec. 3.4, we simply sum and normalize the

cross-modality features for aggregation. Meanwhile, we have also attempted to make this module learnable by concatenating and feeding the features to several fully-connected layers. However, the experimental results showed a considerable performance decrease from such choice, i.e., 2 points decrease on HPatches even compared with the base model, GeoDesc. Our observation is that the raw local features are supposed to be preserved as much as possible, and a learnable aggregation would result in over-parameterization and inability to represent the local detail.

A.2 Training with softmax temperature

We plot the growth of softmax temperature and its respective loss decrease in Fig. 7. As can be seen, the softmax temperature fast grows at the beginning and gradually converges to a constant value, $\sim$38. As mentioned in Sec. 3.5, the softmax temperature is regularized with the same network weight decay, whereas we have observed that eschewing the regularization does not harm the performance, but results in a larger temperature value, i.e., $\sim$42.

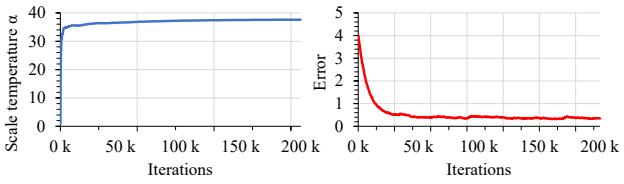

Figure 7: Left: the growth of scale temperature. Right: the respective decrease of loss.

A.3 Ratio test

In previous experiments on image matching, we did not apply any outlier rejection (e.g., mutual check, ratio test [22]) for all methods for fair comparison, whereas the early outlier rejection is critical and necessary to later geometry computation, e.g., recovering camera pose. In particular, ratio test [22] has demonstrated remarkable success, we thus follow the practice in [23] to determine the ratio criteria of the proposed augmented descriptor. Specifically, given *# Correct Matches* defined in Sec. 4.3, we test on HPSequence [2] and aim to find a proper ratio that achieves *Precision = # Putative Matches / # Correct Matches* similar to SIFT. As a result, we choose 0.89 for the proposed descriptor.

To demonstrate the efficacy of the obtained ratio, we evaluate on the wild indoor/outdoor data [47, 42] with an error metric of relative camera pose accuracy. Following the protocols defined in [49], we use mean average precision (mAP) of a certain threshold (e.g., 20°) to quantify the error of rotation and translation. For comparison, we use ratio criteria of 0.80 for SIFT [22] and 0.89 for GeoDesc [23],

	SIFT[22]	**GeoDesc [23]**	**Ours**
ratio criteria	0.80	0.89	0.89
mAP of pose (error threshold 20°)			
indoor	37.4	41.8	**42.9**
outdoor	17.9	20.5	**22.5**

Table 7: Pose evaluation on wild datasets with ratio test applied: *indoor* SUN3D [47] and *outdoor* YFCC100M [42].

and present evaluation results in Tab. 7, which demonstrates consistent improvements with proper outlier rejection.

A.4 Different domain sizes

Somewhat counter-intuitively, the CS structure improves marginally on image matching tasks as reported in Tab. 1. To further study this phenomenon, we compare the patch sampling from different domain sizes, including the original SIFT's ($1\times$) as used in previous experiments, half ($0.5\times$) or double ($2\times$) sizes. We also compare the aggregation of multiple sizes, i.e., the original and halved $(1+0.5)\times$ or the original and doubled $(1+2)\times$. Instead of concatenating features as used by CS structure, we apply simple summing-and-normalizing aggregation in Sec. 3.4 to avoid increasing the dimensionality.

We experiments with our *ContextDesc+* model in Tab. 1, and present the comparison results with different domain sizes in Tab. 8. As can be seen, when only single size is adopted, the original '$1\times$' performs best as being consistent with the training. In addition, when combining a larger size ($(1+2)\times$), we can further boost the proposed method by a considerable margin, yet leading to excessive computational cost and doubling the inference time. In practice, the aggregation with different domain sizes is compatible with the proposed framework, and can be applicable when high accuracy is in demand.

domain size	**Recall *i/v***	
$0.5\times$	61.59	69.79
$2\times$	62.84	71.86
$1\times$ (**ContextDesc+**)	**67.14**	**76.42**
$(1+0.5)\times$	67.31	76.16
$(1+2)\times$	**67.74**	**77.51**

Table 8: The efficacy of extracting local features from different domain sizes.

A.5 Invariance to density change

We further demonstrate the robustness regarding density change on HPSequences [2], of which images are feature-rich and have keypoints up to 15k. Beside of sampling keypoints of different numbers, we consider a more challenging

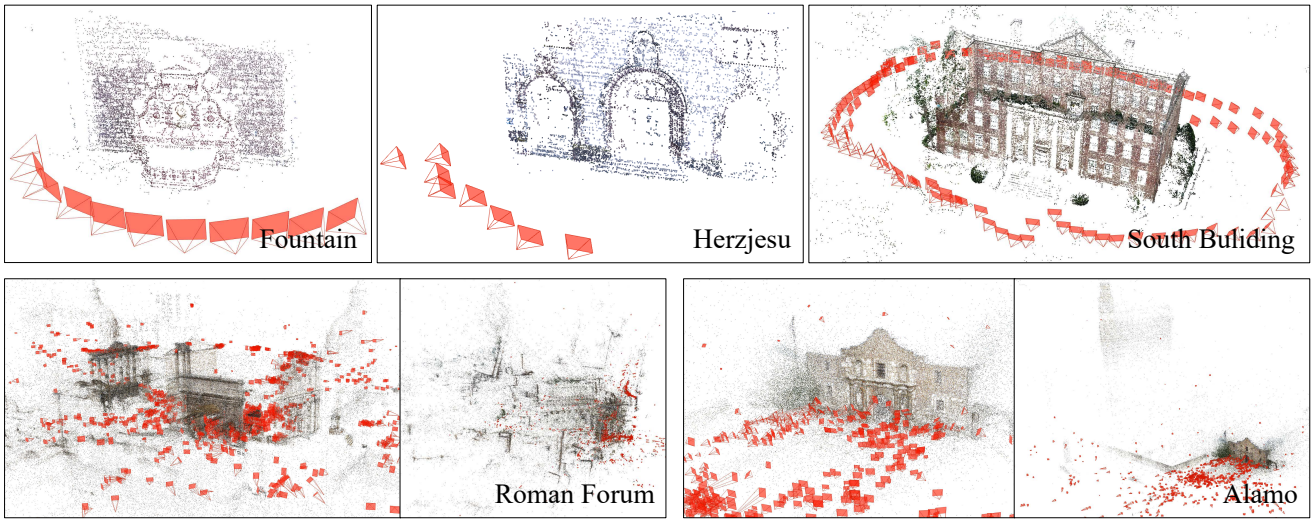

Figure 8: Visualizations of SfM results of Sec. 4.5 from the proposed augmented feature.

case where *all detected keypoints* are used. As presented in Fig. 9, the proposed method delivers consistent improvements in terms of all cases, which demonstrates the reliable invariance property acquired by context encoders.

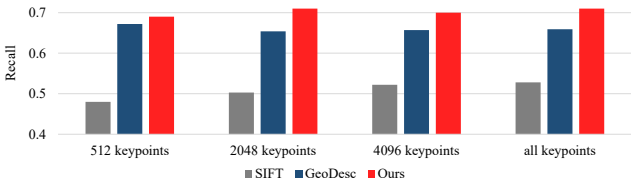

Figure 9: The performance of proposed augmentation scheme regarding density change of keypoints.

A.6. Efficacy of the matchability predictor

To better interpret the functionality of the proposed matchability predictor in Sec. 3.2, we quantitatively evaluate its performance being used as a keypoint detector. Following [35], we apply the matchability predictor onto the entire image, then select 2048 top responses after NMS as keypoints. whose performance is measured by *Repeatability*. Compared with SIFT detector, the results are improved from **32.81 to 37.93** and **25.53 to 26.34** on *i/v* sequences of HPatches. While the detector performance is not the focus of this paper, we believe that by adopting more advanced techniques, this module will potentially benefit to the *joint training* of keypoint detector and descriptor, and have large rooms for future improvements.

A.7 Application on image retrieval

We use an open-source implementation of VocabTree[1] [39] for evaluation image retrieval performance, and compare SIFT [22], GeoDesc [23] and the proposed ContextDesc. The mAPs on Paris dataset [30] from three competitors are **49.89, 53.84 and 61.29**, while on Oxford buildings [29] are **47.27, 53.29 and 61.64**. By re-ranking the top-100 by spatial verification [29], the mAPs on Paris are improved to **52.23, 55.02 and 64.53**, while on Oxford are **51.64 , 54.98 and 65.03**. The experimental results effectively demonstrate the superiority of the proposed method.

A.8 More visualizations

We have provided more visualizations of previous experiments in Fig. 8 (SfM results in Sec. 4.5) and Fig. 10 (image matching results w.r.t different image transformations).

[1] https://github.com/hlzz/libvot

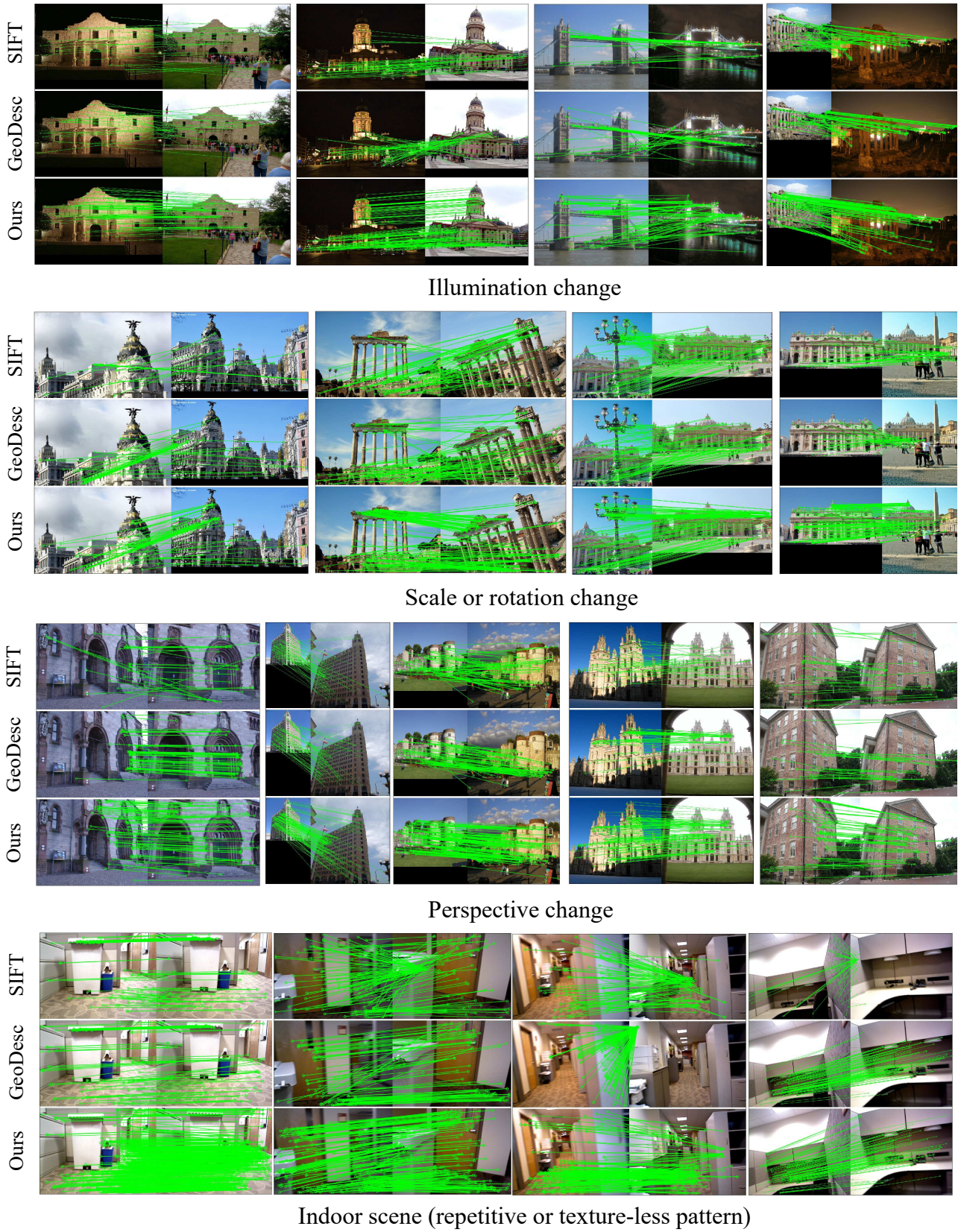

Figure 10: Image matching results after RANSAC. From top to bottom: SIFT, GeoDesc and proposed augmented feature.