1 Appendix

1.1 Self-labeling

In our experiments, the total iteration steps are set to 2. Since the detection threshold is crucial to the quality of the pseudo-ground truth label, we deploy the following strategy to find the proper threshold for different step: For a base detector, we randomly choose 200 COCO images and use different thresholds to label them, which shows that the detection thresholds are stable in the range [0.001, 0.005] and [0.015, 0.030] for each step, respectively. Then we empirically choose a threshold that gives the best label effects.

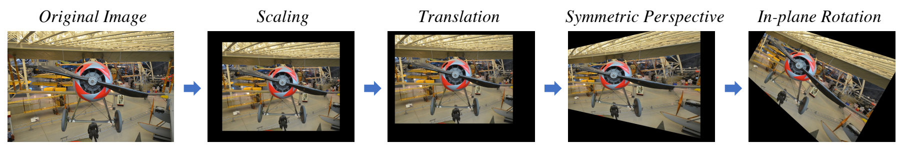

Fig. 1. The visualization of homography sampling. The input image is sequentially transformed by Scaling, Translation, Symmetric Perspective, and In-plane Rotation.

1.2 Homography sampling parameters

As stated in the paper, during the training, each image I in COCO is transformed by a randomly sampled homography to synthesize the corresponding image I', resulting in the image pair. Like SuperPoint[1], the sampled homography combines four simple transformations, namely scaling, translation, symmetric perspective, and in-plane rotation. To ensure the sampled homography is reasonable, we constraint these sub-transformations in the following range:

$$\begin{aligned} Scaling:&\ [0.8, 2.0], & Translation:&\ [-0.1, 0.1], \\ Symmetric\ Perspective:&\ [-0.3, 0.3], & In\text{-}plane\ Rotation:&\ [-\pi/2, \pi/2], \end{aligned}$$

where the sampled value of *Scaling*, *Translation*, and *Symmetric Perspective* is relative to the input image's spatial size. The process of homography sampling can be seen in Fig. 1.

1.3 Photometric augmentation parameters

During the training, the same as SuperPoint[1], we use photometric augmentation to strengthen the model's robustness. Before an image input to the model for the training, it will be randomly processed by a series of sub-augmentations: 1) *Brightness*: Randomly adds value to all pixels; 2) *Contrast*: Randomly adjusts

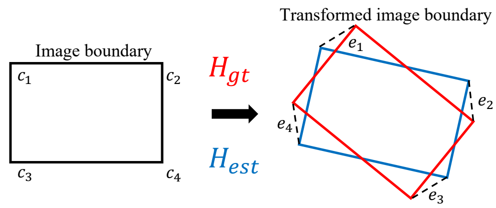

Fig. 2. The computation of homography error(HE). It is the mean distance between corners of the target image after being transformed by 1) the ground truth homography H_{gt} and 2) the estimated homography H_{est}. The dashed line e_i denotes the error.

image contrast by a scale; 3) *Gaussian Noise*: Randomly adds noise sampled from Gaussian distributions; 4) *Impulse Noise*: Randomly adds impulse noise. 5) *Motion Blur*: Randomly blurs an image with a given probability. The parameters of these sub-augmentations are listed as follows:

$$\begin{aligned} Brightness &: [-50, 50], & Contrast &: [0.5, 1.5], \\ Gaussian\ Noise &: \mu = 0, std \in [0, 10], & Impulse\ Noise &: [0, 0.0035], \\ Motion\ Blur &: p = 0.5, kernel = 3. \end{aligned}$$

For *Gaussian Noise*, the operation samples an *std* from the given range and generates Gaussian noise based on this *std*. Similarly, *Impluse Noise* samples a probability p and produces the noise under this probability.

1.4 Computation of homography accuracy

The homography accuracy(HA) on HPatches is evaluated based on the homography error(HE). First, given a ground-truth homography transformation H_{gt} and the estimated one H_{est}, the HE is computed as follows:

$$HE = \frac{1}{4}\sum_{i}^{4}||(H_{gt} - H_{est})c_i||, \tag{1}$$

where c_i is the ith corner of the original image, and the process is shown in Fig. 2. Then the homography accuracy under a threshold ϵ(1-10 used in the paper) can be formulated as:

$$HA = \frac{1}{n}\sum_{i}^{n}(HE_i <= \epsilon). \tag{2}$$

1.5 Computation of recall

To compute $\%Recall$ on FM-Bench, the average of normalized symmetric epipolar distance is used. This metric's detailed computation is illustrated in Alg.1

Algorithm 1: Average of Normalized Symmetric Geometry Distance

Input : $F_1, F_2, N, h_1, w_1, h_2, w_2, I_1, I_2$
Output: $nsgd$

```
1  nsgd = 0
2  count = 0
3  while count < N do
4      randomly choose a point m in I_1
5      draw l_1 = F_1 m in I_2
6      if the epipolar line doesn't intersect in I_2 then
7          go back to step 4
8      end
9      randomly choose a point m' in l_1
10     draw l_2 = F_2 m in I_2
11     d' = distance(m', l_2) / sqrt(h_2^2 + w_2^2)
12     draw l_3 = F_2^T m' in I_1
13     d = distance(m, l_3) / sqrt(h_1^2 + w_1^2)
14     nsgd = d' + d
15     count = count + 1
16 end
17 swap(F_1, F_2)
18 repeat step 2-15
19 ansgd = nsgd/4N
20 return ansgd
```

and Fig. 3, where I_1, I_2 are the input image pair, and $F1, F2$ are the ground-truth fundamental matrix and the estimated fundamental matrix, respectively. Given $ansgd$, one can evaluate $\%Recall$ under a threshold β(0.05 as default[2]) as follows:

$$Recall = \frac{1}{n}\sum_{i=0}^{n}(ansgd_i <= \beta). \tag{3}$$

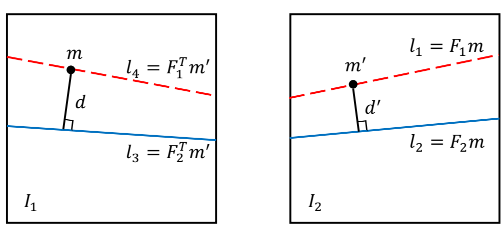

Fig. 3. Visualization of the epipolar distance between two fundamental matrices. Given m in I_1, one can generate epipolar line l_1 based on F_1, and epipolar line l_2 based on F_2. Analogously, l_3 and l_4 is the epipolar lines of m' respectively. The epipolar distance is thus defined as m' to l_2, and m to l_3.

1.6 More visualization results

Here we give more qualitative detecting and matching samples of our MLIFeat, which is shown in Fig. 4.

References

1. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2018) 224–236
2. Bian, J.W., Wu, Y.H., Zhao, J., Liu, Y., Zhang, L., Cheng, M.M., Reid, I.: An evaluation of feature matchers for fundamental matrix estimation. arXiv preprint arXiv:1908.09474 (2019)
3. Sattler, T., Weyand, T., Leibe, B., Kobbelt, L.: Image retrieval for image-based localization revisited. In: BMVC. Volume 1. (2012) 4
4. Balntas, V., Lenc, K., Vedaldi, A., Mikolajczyk, K.: Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 5173–5182

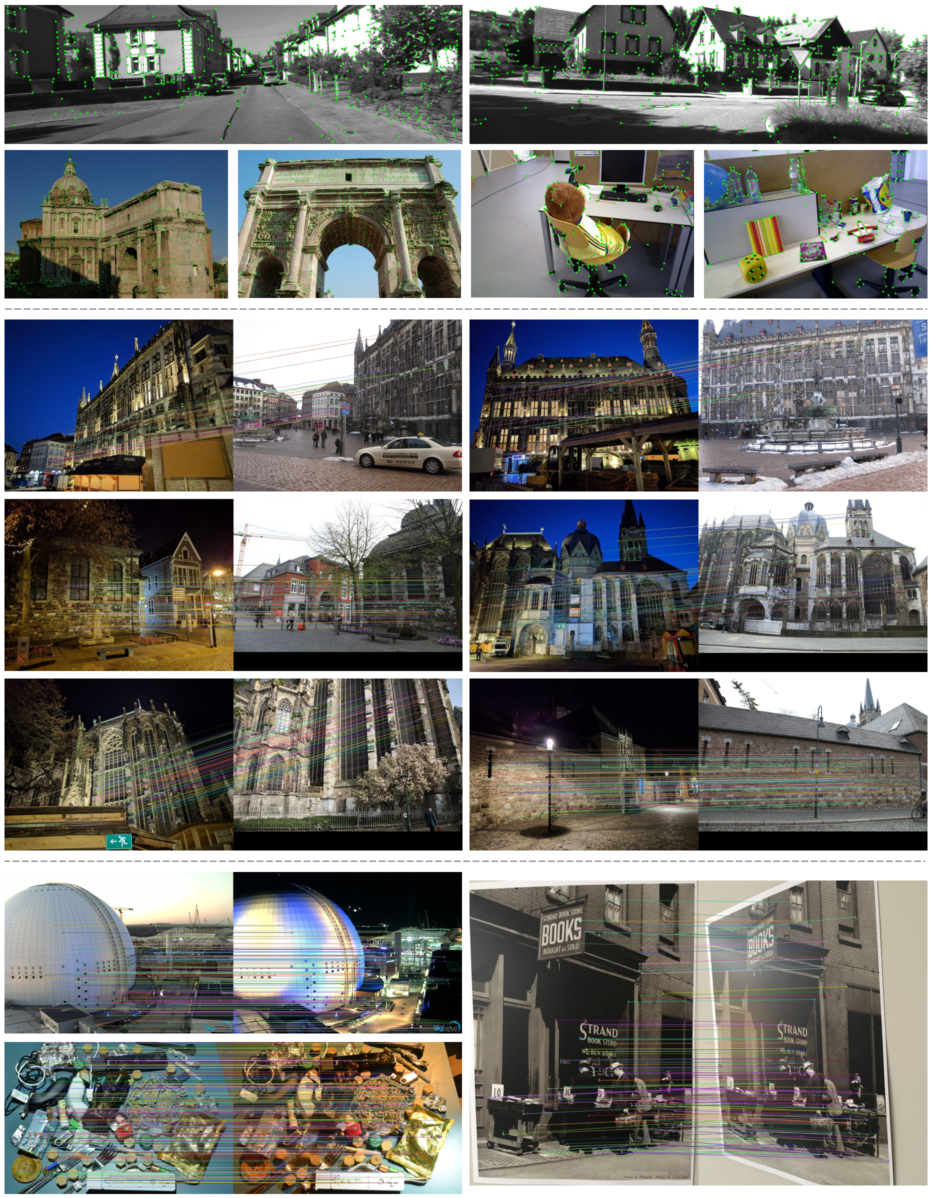

Fig. 4. Extra visualization samples of detecting and matching. The top block contains the detection samples of FM-Bench[2]. The middle and the bottom block contains the matching samples of Aachen-Day-Night[3] and HPatches[4], respectively.