This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Adaptive Assignment for Geometry Aware Local Feature Matching

Dihe Huang[1*] Ying Chen[2*] Yong Liu[2] Jianlin Liu[2] Shang Xu[2]
Wenlong Wu[2] Yikang Ding[1] Fan Tang[4†] Chengjie Wang[2,3†]
[1]Tsinghua Univeristy [2]Tencent YouTu Lab [3] Shanghai Jiao Tong University
[4] Institute of Computing Technology, Chinese Academy of Sciences
{hdh20, dyk20}@mails.tsinghua.edu.cn tfan.108@gmail.com
{mumuychen, choasliu, jenningsliu, shangxu, wenlongwu, jasoncjwang}@tencent.com

Abstract

*The detector-free feature matching approaches are currently attracting great attention thanks to their excellent performance. However, these methods still struggle at large-scale and viewpoint variations, due to the geometric inconsistency resulting from the application of the mutual nearest neighbour criterion (*i.e.*, one-to-one assignment) in patch-level matching. Accordingly, we introduce AdaMatcher, which first accomplishes the feature correlation and co-visible area estimation through an elaborate feature interaction module, then performs adaptive assignment on patch-level matching while estimating the scales between images, and finally refines the co-visible matches through scale alignment and sub-pixel regression module. Extensive experiments show that AdaMatcher outperforms solid baselines and achieves state-of-the-art results on many downstream tasks. Additionally, the adaptive assignment and sub-pixel refinement module can be used as a refinement network for other matching methods, such as SuperGlue, to boost their performance further. The code will be publicly available at https://github.com/AbyssGaze/AdaMatcher.*

1. Introduction

Establishing accurate correspondences for local features between image pairs is an essential basis for a broad range of computer vision tasks, including visual localization, structure from motion (SfM), simultaneous localization and mapping (SLAM), etc. However, achieving reliable and accurate feature matching is still challenging due to various factors such as scale changes, viewpoint diversification, illumination variations, repetitive patterns, and poor texture.

Existing image matching pipelines are mainly divided into two types: detector-based and detector-free. The former is to build matches on detected and described sparse keypoints [8, 19, 20, 23, 26, 32]. However, as the detector-based matching pipeline relies on the reliability of keypoint detectors and features description, it tends to perform poorly under large viewpoint changes or scale variations. For the latter, the detector-free matching pipeline can take full advantage of the rich context to establish corre-

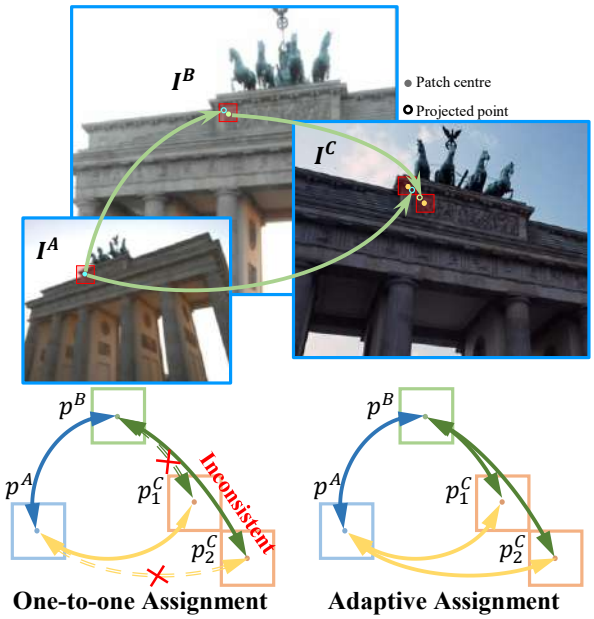

Figure 1. **An illustration of one-to-one assignment and adaptive assignment.** Under viewpoint changes or scale variations, one-to-one assignment leads to geometric inconsistency in patch-level feature matching, while adaptive assignment does not. For example, with one-to-one assignment, patch pair (p^A, p_2^C) is treated as a negative example, even though both p_1^C and p_2^C are projected into p^A of I^A. Such a matching rule is inconsistent with two-view and multi-view projective geometry.

*These authors contributed equally.
†Corresponding author.

spondence between images end-to-end [6,13,24,25,29–31], without independent keypoint detection and feature description steps. To achieve efficiency and accurate matching, the SOTA detector-free matching pipelines [6,11,13,29,30] use a coarse-to-fine structure, in which the patch-level matches are first obtained using the mutual nearest neighbor criterion, and then are refined to a sub-pixel level.

Although these methods have improved considerably in performance, they still perform unsatisfactorily in extreme cases (*e.g.*, large viewpoint changes and scale changes). This is due to the fact that applying the mutual nearest neighbor criterion (ie, one-to-one correspondence) in patch-level matching leads to geometric inconsistencies and difficulties in obtaining sufficient high-quality matches under large-scale or viewpoint variations. As shown in Fig.1, where I^A, I^B, I^C are from the same scene, p_1^C and p_2^C of I^C are both projected into p_A of I^A. However, when the mutual nearest neighbour criterion is applied in the training process, the patch pair (p^A, p_1^C) is treated as a positive sample, while the patch pair (p^A, p_2^C) is treated as a negative sample. The incorrect assignment leads to two-view geometric inconsistency. Deeply, from a multi-view perspective, (p^A, p^B) and (p^B, p_2^C) are positive samples while (p^A, p_2^C) is a negative sample, which leads to multi-view geometric inconsistency between multiple image pairs. For inference, when there are large viewpoint changes or scale variations, one-to-one matching is difficult to obtain enough inliers to ensure accurate camera pose estimation. Furthermore, when applied to multi-view-based downstream tasks (*e.g.*, SfM and 3D reconstruction), one-to-one patch-level correspondences do not guarantee the consistency of multi-view matching, which is likely to make the mapping fail or the bundle adjustment difficult to converge.

Inspired by the above consideration, we present AdaMatcher, a geometry aware local feature matching approach, targeting at mitigating potential geometry mismatch between image pairs without scale-alignment preprocessing or viewpoint warping. Different from dual-softmax or optimal transport in [28, 29] which guarantees one-to-one correspondence, we allow adaptive assignment (including many-to-one and one-to-one) at patch-level matching during training and inference. When the scale or viewpoint changes significantly, the adaptive assignment can guarantee matching accuracy. The smooth scale transition from many-to-one matches between image pairs can be adopted to resolve scale inconsistencies. Furthermore, the structure of our delicately designed feature interaction module couples co-visible feature decoding with cross-feature interaction, allowing the probability map of the co-visible region to be obtained later by a simple module to filter out matches outside co-visible areas. To summarize, we aim to provide several critical insights of matching local features across scales and viewpoints:

- We propose a detector-free matching approach AdaMatcher that allows a patch-level adaptive assignment followed by a sub-pixel refinement to guarantee the establishment of geometry aware feature correspondences.
- We introduce a novel feature interaction structure, which couples the co-visible feature decoding and cross-feature interaction. The probability map of the co-visible area can be obtained later by an additional module.
- Extensive experiments and analysis demonstrate that AdaMatcher outperforms various strong baselines and achieves SOTA results for many downstream vision tasks.

2. Related Work

2.1. Scale- or Viewpoint-invariant Local Feature

To tackle geometry deformation induced by scale and viewpoint variation across images, tremendous efforts [2–4, 7, 8, 18–20, 23, 26] have been made within local feature matching pipelines. Hand-crafted local features such as SIFT [19] or ORB [26] adopt scale-space theory [17] to alleviate potential large-scale variations. However, descriptors extracted locally from low-level image textures possess poor discrimination ability. Recently, many works have been devoted to a learning-based approach to tackle local feature matching under scale variations or viewpoint changes. Methods directly performing convolution upon the multi-scale pyramid such as KeyNet [2], R2D2 [23] and HDDNet [4], or implicitly applying multi-scale detection such as ASLfeat [20] and DenseNet [18] are intended to mimic conventional scale space theory. However, the multi-scale pyramid brings the side effect of ambiguity since correspondence needs to be established among multiple scale levels. There are also works [8, 22] aiming at invariance to different scales through an elaborate learning process, however, which would render them less discriminative [34]. Some works [5,35] in geo-localization aim to achieve viewpoint invariance. GeoWarp [5] directly warps pairwise images to a closer geometrical space to eliminate viewpoint inconsistencies and then computes their similarities using dense local features for image retrieval tasks. In addition, OETR [7] estimates overlap areas as a preprocessing module in the existing detector-based matching pipeline to constrain keypoint detection in the co-visible areas and eliminate scale and viewpoint inconsistencies. However, its need to scale up the whole image increases the time and computational consumption of the later feature extraction and matching steps. In contrast to the above approaches, our proposed method is inspired by many-to-one matching caused by viewpoint and scale variations. The adaptive as-

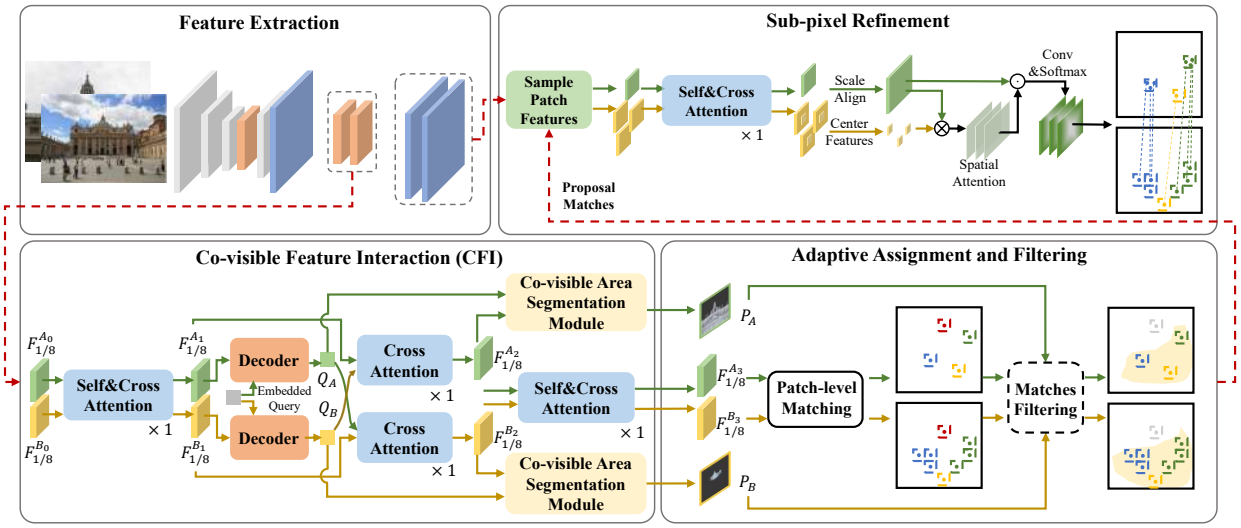

Figure 2. **Architecture of AdaMatcher.** A local feature CNN extracts two feature maps with size of 1/2 and 1/8 of the input image dimension. Afterwards, 1/8 size features of the two images are correlated by our feature interaction module, followed by an extra module to estimate the co-visible area shared between two images. Then the interacted patch features and co-visible probability maps are fed into an adaptive assignment and filtering module, yielding sufficient patch-level matches. Finally, proposal-matched features are sampled in the 1/2 size feature maps with a scale alignment to mitigate scale mismatch, followed by sub-pixel refinement.

signment is introduced to handle feature matching under view and scale variations, and the relative scales between two images can be estimated from the results of patch-level adaptive matching, which can be used for feature scale alignment in the subsequent refinement step. In addition, inspired by the overlap estimation of OETR [7], we couple the co-visible feature decoding into feature interaction to make the network more focused on the co-visible regions, thus alleviating the performance degradation in extreme cases.

2.2. Detector-free Image Matching

Recent works [6, 13, 25, 29–31] have shown us that end-to-end dense feature matching without keypoint detection can be more robust than detector-based matching methods in many scenarios. NCNet [25] and its follow-ups [13, 24] propose a 4D matching cost volume to enumerate all possible correspondences and obtain dense matches end-to-end. Although all the potential matches are considered in the 4D matching tensor, the receptive field of 4D convolution is still limited to each match's neighborhood area. Benefiting from the global receptive field and long-range dependencies from Transformers, LoFTR [29] and its variants [6, 30] extend neighborhood consensus to the whole image, setting the SOTA performance for dense feature matching approaches. However, [6, 13, 24, 25, 29–31] do not handle the case of significant viewpoint and scale changes well because they follow one-to-one matching. In comparison, the use of adaptive assignment can make the dense feature matching methods more robust in extreme cases.

3. Methods

This section describes our proposed matching framework, named AdaMatcher, as shown in Fig.2. Given a pair of images I^A and I^B, we first feed them into a CNN backbone to obtain coarse features and fine features. Then, the coarse features are passed through our CFI module (Sec. 3.1) to accomplish feature Interaction and co-visible area estimation. After that, adaptive assignment (Sec.3.2) is applied to get the patch-level matches and calculate the relative scales between image pairs, while the previously estimated co-visible regions are used to filter the matches. Finally, the patch-level matches are scale-aligned and refined to sub-pixel precision (Sec. 3.3) according to the estimated scale.

3.1. Co-visible Feature Interaction

Our ultimate goal is to bring existing detector-free matching methods to be more robust under scale or viewpoint changes. We find that overlap estimation [7] helps to improve the matching performance in extreme cases. However, introducing a full network for co-visible region estimation would be computationally and time-consuming. Instead, we couple feature interactions with co-visible feature decoding so that co-visible features can be used to guide global feature interactions while reducing computation. On the one hand, co-visibility guidance can suppress features in non-co-visible regions, facilitating the subsequent matching step. On the other hand, a simple additional module can be used to obtain co-visible regions to filter mismatches.

Feature Interaction with Co-visible Feature Decoding. As shown in the bottom left part of Fig.2, we first use one set of self- and cross-attention layer as the feature encoder to acquire information within and across images. The output features are denoted as $F_{1/8}^{A_1}$ and $F_{1/8}^{B_1}$ respectively. For co-visible feature initialization, we adopt one query $Q \in \mathbb{R}^{1\times d}$ to embed co-visible context, where d is the channel dimension, and perform one cross-attention layer to decode the locations of the co-visible region. Q^A and Q^B denote the features decoded from $F_{1/8}^{A_1}$ and $F_{1/8}^{B_1}$ respectively, i.e., $Q^i = \mathbf{transfomer}(q = Q, k = v = F_{1/8}^{i_1}), i \in \{A, B\}$. After that, co-visible features can be used to guide global feature interactions through a cross transformer, of which the outputs are denoted as $F_{1/8}^{A_2}$ and $F_{1/8}^{B_2}$ respectively. Finally, to make the local features more distinguishable, we use another set of self- & cross-attention layers to construct a complete graph for feature correlation. The proposed feature interaction structure can be directly applied to LoFTR [29], QuadTree [30] and ASpanFormer [6], *i.e.*, the corresponding variants can be obtained according to different attention mechanisms.

Co-visible area segmentation. After co-visible feature decoding, a simple additional module can be used to obtain the co-viewing regions. Here we consider co-visible area segmentation as predicting a logit probability map, whose values at each pixel represent the probability of being in the co-visible region, as shown in Fig.3. In detail, we project the decoded features $Q^{A,B}$ to construct a weight map using matrix multiplication and a sigmoid function. The weight map is used to enhance the co-visible context in feature maps $F_{1/8}^{A_2}, F_{1/8}^{B_2}$. Then a convolution operation with the kernel size of 3×3 and a sigmoid function are applied, which is detailed in Eq. (1):

$$\begin{aligned} \mathbf{weight_i} &= \mathbf{Sigmoid}((F_{1/8}^{i_2})^T Q^i), \\ P_i &= \mathbf{Sigmoid}(\mathbf{Conv}(\mathbf{weight_i} \odot F_{1/8}^{i_2} + F_{1/8}^{i_2})), \end{aligned} \tag{1}$$

where $i \in \{A, B\}$, $\odot$ denotes element-wise multiplication and P_A, P_B denote the co-visible probability map of image A and image B. After obtaining P_A and P_B, a confidence threshold can be applied to retain the co-visible areas in image A and image B.

3.2. Adaptive Assignment

In this section, we will elaborate on one of the main contributions of our work: adaptive assignment when matching between features across images. As shown in Fig.4(a), when scale varies or viewpoint changes, centers of several patches within one image would be projected into only one patch of the other image, named many-to-one correspondences. For the ground truth patch-level labels obtained by one-to-one assignment, only the correspondences that satisfy the mutual nearest neighbors constraint are taken as positive samples, while the others as negative samples. Such ambiguous label assignment is detrimental to supervised training. As shown in Fig.4(b), while a set of patch centers (or pixels) of image A: $\{P_i^A | p_{ik}^A, k = 1, 2, ..., N\}$ are all projected into a patch (or pixel) of image B: p_j^B using ground-truth camera poses and depth maps, features corresponding to $\{P_i^A\}$ are similar to the feature corresponding to p_j^B. Following mutual nearest neighbors constraint, (p_{im}^A, p_j^B) would be assigned as positive sample, while $\{(p_{ik}^A, p_j^B) | k = 1, 2, ..., N, k! = m\}$ are assigned as negative, where $m = \arg\min_k \|\mathrm{D}(\mathrm{W}(p_{ik}^A), p_j^B)\|$, $\mathrm{D}(\cdot)$ is the projected distance between matching candidates and $\mathrm{W}(\cdot)$ demonstrates the projection function. Such one-to-one assignment criterion will turn good correspondences into negative samples, which is inconsistent with the multi-view geometry theory. Instead, adaptive assignment will make correspondences $\{(p_{ik}^A, p_j^B) | k = 1, 2, ..., n\}$ being positive samples since their appearances are similar and they also conform to geometric constraint. When applied to multi-view tasks (*e.g.* SfM), the one-to-one assignment cannot guarantee the multi-view geometric consistency. Instead, when adaptive many-to-one/one-to-many/one-to-one assignments are allowed at patch-level matching, feature inconsistency under large-scale or viewpoint changes will be mitigated.

Figure 3. **Co-visible area segmentation visualizations.** Visualized with two different scenes, the first column is the origin image pair, and the second is the co-visible area.

Matching matrix formulation. Given features $F_{1/8}^{A_3}$ and $F_{1/8}^{B_3}$ output from CFI module, we calculate their similarity matrix $\mathcal{S}$:

$$\mathcal{S}(i, j) = \frac{1}{r} \cdot \left\langle F_{1/8}^{A_3}(i), F_{1/8}^{B_3}(j) \right\rangle, \tag{2}$$

where i and j are index of feature patches in I^A and I^B respectively, and $\langle \cdot, \cdot \rangle$ denotes inner product. Adaptive assignment is a one-way operation that consists of many-to-one and one-to-one assignment, i.e., we assign "many" patches on images with a large co-visible area to "one" patch on images with a small co-visible area. When little scale or viewpoint variation exists, many-to-one assignment is adaptive to become one-to-one. Hence, we apply softmax operation to similarity matrix $\mathcal{S}(i, j)$ on two dimensions separately, followed by selecting similar matches

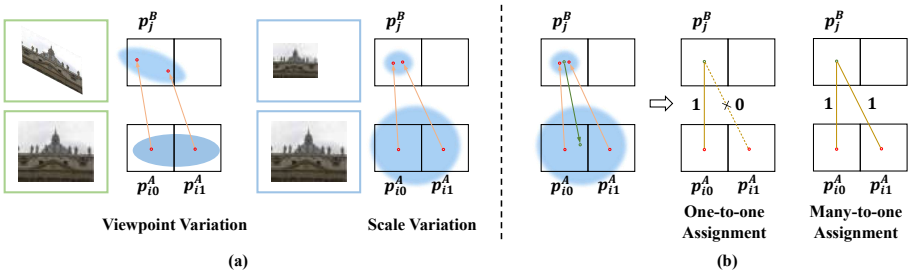

Figure 4. **Comparison of one-to-one and many-to-one assignment.** (a) shows patch-level many-to-one matching due to viewpoint and scale changes; (b) shows the difference between many-to-one and one-to-one assignment: one-to-one only keeps a single match while both p^A_{i0} and p^A_{i1} correspond to p^B_j, and many-to-one assignment keeps common area matches to disambiguate positive and negative samples, to resolve geometry deformation.

with a threshold θ_m:

$$\begin{aligned} \mathcal{P}_k &= \mathbf{softmax}(\mathcal{S}(i,\cdot))_j, \\ \mathcal{M}_k &= \{(\widetilde{i},\widetilde{j}) | \mathcal{P}_k(\widetilde{i},\widetilde{j}) > \theta_m\}, \end{aligned} \tag{3}$$

where $k \in 0, 1$, $\mathcal{P}_0$ and $\mathcal{P}_1$ are the matching probability matrix obtained by softmax operation along the first dimension and the zeroth dimension, $\mathcal{M}_0$ and $\mathcal{M}_1$ are the corresponding patch-level match proposals. Then we select the matching probability matrix $\mathcal{P}$ and the proposal matches $\mathcal{M}$: $\mathcal{P} = \mathcal{P}_{index}, \mathcal{M} = \mathbf{Filtering}(\mathcal{M}_{index}, P_A, P_B, \theta_{co-visible})$, where **Filtering** is to filter out matches outside predicted co-visible areas, $\theta_{co-visible}$ is used to select patches in the co-visible probability maps belong to the co-visible region, and $index = \arg\max_k \{s_k | k = 0, 1\}$. s_k is the scale between images, calculated by

$$s_k = \frac{\mathbf{len}(\mathcal{M}_k)}{\mathbf{len}(\mathbf{unique}(\mathcal{M}_k[:, 1-k]))}, k = 0, 1. \tag{4}$$

3.3. Sub-pixel Refinement Module

By adaptive assignment we obtain patch-level match proposals through scales and viewpoints. Then we refine these match proposals to more accurate sub-pixel level matches, by a scale-alignment and an expectation regression module.

Scale-alignment. Suppose scale for p^A_i is larger than p^B_j, match proposals are $\{p^A_{i\in\Omega}, p^B_j\}$, and Ω is the collection of assigned patches. Patch features can be then sampled in $F^A_{1/2}$ and $F^B_{1/2}$, followed by one self/cross attention layer to communicate feature messages from assigned patches. The scale ratio is $s = max(s_0/s_1, s_1/s_0)$, where s_0 and s_1 could be calculated from Eq. (4). Smaller scale image features are upsampled by this s to compensate for scale mismatch between images.

Sub-Pixel level regression. To locate accurate sub-pixel level matches, we generate a heatmap representing the matching probability for each pixel. Firstly, we correlate center features of $F^A_{i\in\Omega}$ and the scale-alignment features F'^B_j to calculate n spatial attention maps. Then we perform a dot-product operation on the attention maps and F'^B_j to balance the relevance for each features. Finally, a simple convolution and softmax is employed to predict probability distribution, from which the final position i' with sub-pixel accuracy on I_A is obtained by taking expectation over distribution. By this scale-aware adaption to match position refinement, we achieve more accurate sub-pixel matches.

3.4. Supervision

Co-visible Area Segmentation Loss. For the co-visible area segmentation, we treat it as a per-pixel binary classification task. The loss $\mathcal{L}_{co-visible}$ can be calculated by Focal Loss [16] (abbr. as FL hereafter):

$$\mathcal{L}_{co-visible} = FL(P_A, \widehat{P}_A) + FL(P_B, \widehat{P}_B), \tag{5}$$

where $\widehat{P}_A$ and $\widehat{P}_B$ denote the ground-truth co-visible areas of image A and image B, respectively, which are calculated based on depth and camera poses.

Proposal Matching Loss. For the loss function of the adaptive matching probability matrix $\mathcal{P}$, we use the same Focal Loss [16] as in LoFTR [29]:

$$\mathcal{L}_M = FL(\mathcal{P}, \widetilde{\mathcal{P}}), \tag{6}$$

where $\widetilde{\mathcal{P}}$ is the ground-truth labels of the adaptive matching probability matrix calculated from the camera poses and depth maps, and $\mathcal{P}$ is the predicted matching probability matrix. For the α and γ parameters in Focal Loss [16], we use the default values, which are set to 0.25 and 2, respectively. For patch-level label generation, we project the patch centroids of the two images onto each other using depth and camera poses and then characterize all points projected into the same patch as positive samples with that patch.

Refinement Loss. Inspired by LoFTR [29], we use the same loss $\mathcal{L}_{refine} = \frac{1}{|\mathcal{M}^k_{gt}|} \sum_{i,j'} \frac{1}{\sigma(i)^2} \left\| j' - j'_{gt} \right\|$ function for the final predicted matches, where k is the $index$ calculated above. M^k_{gt} is the ground-truth matches calculated

from ground-truth depths and camera poses and $\sigma^2(\cdot)$ is the variance of the corresponding heatmap [29].

Our final loss is balanced as:

$$\mathcal{L} = 0.5 * \mathcal{L}_{co-visible} + 1.0 * \mathcal{L}_{M} + 1.0 * \mathcal{L}_{refine}. \quad (7)$$

3.5. Implementations

We train AdaMatcher on the MegaDepth datasets following [29], without any data augmentation. We apply AdaMatcher to LoFTR [29] and its variants (QuadTree Attention [30] and ASpanFormer [6]), named AdaMatcher-LoFTR, AdaMatcher-Quad and AdaMatcher-ASpan, respectively. The only difference between these three variants is the type of attention mechanism in the feature interaction module, which are linear attention [12], quadtree attention [30] and attention span [6]. All networks are trained using AdamW optimizer with initial learning rate of 8×10^{-3} and batch size of 8. It converges after 2 days of training on 8 V100 GPUs. The image feature extractor is a standard ResNet-FPN [10, 15] architecture, which is identical to LoFTR [29]. θ_m is set to 0.5, $\theta_{co-visible}$ is set to 0.2 and the patch window size w for refinement is set to 5. The number of channels of the $F_{1/8}$ and $F_{1/2}$ is 256 and 128, respectively. To save GPU memory usage during training, we sample 30 percent of matches (max to 2500) from the match proposals for supervision in sub-pixel refinement module. More details are provided in Supplementary Material A.

4. Experiments

4.1. Homography Estimation

Dataset. HPatches [1] is the most widely used image matching evaluation dataset. There are 116 scenes with 57 sequences of large illumination variations and 59 sequences under significant viewpoint changes to evaluate our method under different circumstances. All images are resized to their longer dimensions equal to 1024, and we limit the maximum amount of matches to 1K for all methods.

Metrics. Following [8, 28, 37] we use corner correctness to describe the performance of estimated homography. Four corners in the first reference image are wrapped to the other image by estimated homography. Then percentange of correct estimated homographies whose average error of the four corners is less than $1/3/5$ pixels demonstrates the matching *Accuracy*. We use OpenCV RANSAC as the robust estimator following [37].

Results. We split matching methods into "Detector-based" and "Detector-free" as in LoFTR [29]. Tab.1 shows that AdaMatcher notably performs on par with or better than other baselines under all error thresholds. Under viewpoint variations, many-to-one corresponding is more appropriate, and adaptive assignment eliminates the matching ambiguity, making Adamatcher superior to other methods that use one-to-one assignment.

Category	Method	Overall	Viewpoint
		Accuracy(ϵ <1/3/5px)	
Detector-based	KeyNet [2]+HardNet [21]	0.30 /0.61 /0.75	0.14 /0.46 /0.64
	SIFT [19]+HardNet [21]	0.33 /0.59 /0.74	0.20 /0.40 /0.60
	SP [8]	0.31 /0.66 /0.78	0.18 /0.51 /0.64
	R2D2(MS) [23]	0.29 /0.60 /0.72	0.18 /0.43 /0.58
	SP [8]+CAPS [33]	0.27 /0.66 /0.71	0.15 /0.53 /0.65
	Patch2Pix [37]	0.34 /0.68 /0.79	0.16 /0.47 /0.63
	SP [8]+SG [28]	0.34 /0.67 /0.81	0.21 /0.53 **/0.72**
	SP [8]+SG [28]+Ada	0.35 /0.71 /0.81	0.24 **/0.59 /0.72**
Detector-free	LoFTR-OT [29]	0.41 /0.70 /0.79	0.15 /0.47 /0.61
	LoFTR-DS [29]	0.44 /0.73 /0.82	0.19 /0.54 /0.67
	AdaMatcher-LoFTR	0.49 **/0.75** /0.83	0.26 /0.57 /0.69
	QuadTree [30]	0.48 /0.70 /0.81	0.20 /0.48 /0.65
	AdaMatcher-Quad	0.47 **/0.75** /0.83	0.26 /0.58 /0.69
	ASpanFormer [6]	0.46 /0.72 /0.82	0.22 /0.51 /0.68
	AdaMatcher-ASpan	**0.50 /0.75 /0.84**	**0.27** /0.57 /0.70

Table 1. **Homography estimation on HPatches.** The better methods are underlined, and the best overall method is highlighted in bold. Under viewpoint changes, AdaMatcher has substantial performance improvements compared to the corresponding baselines.

4.2. Relative Pose Estimation

Datasets. We use MegaDepth [14] to demonstrate the effectiveness of AdaMatcher for pose estimation in outdoor scenes. Following [7], we used a scale-split Megadepth test set (with 10 scenes), as scale ratio ranges in $[1, 2)$, $[2, 3)$, $[3, 4)$, $[4, +\infty)$. Fig.5 qualitatively shows the matching result of LoFTR and AdaMatcher in MegaDepth. All images (both training and test) are resized so that the longest dimension equals 840 (ASpanFormer [6] and QuadTree Attention [30] use 832 due to their need for an image resolution divisible by 16).

Metrics. Following [28], we report the **AUC** of the pose error under thresholds $(5^\circ, 10^\circ, 20^\circ)$, where the pose error is set as the maximum angular error of relative rotation and translation. In our evaluation protocol, the relative poses are recovered from the essential matrix, estimated from feature matching with RANSAC.

Comparative methods. We compare AdaMatcher with traditional and current SOTA methods: 1) detector-based methods including SIFT [19]+HardNet [21], KeyNet+HardNet [21], R2D2 [23], ASLFeat [20], Disk [32], SuperGlue(SG) [28] with SuperPoint(SP) [8] or Disk [32] detector and SuperGlue [28] with OETR [7] for pre-processing, 2) detector-free methods including PDC-Net [31], LoFTR [29], QuadTree Attention [30] and ASpanFormer [6].

Results on MegaDepth. When the relative scale ratio is small, AdaMatcher performs slightly better than LoFTR. As the scale difference increases, AdaMatcher outperforms its counterparts more obviously. Though SIFT [19] (even combined with HardNet [21] to more discriminative descriptors) detects keypoint in scale space, R2D2 [23] utilizes multi-

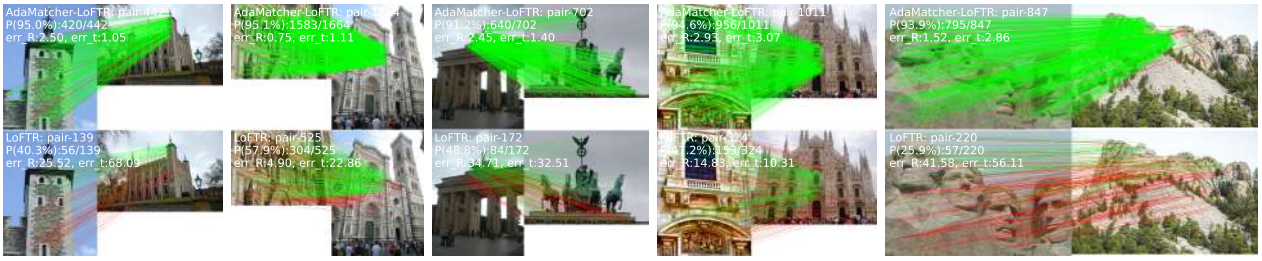

Figure 5. **Qualitative results.** AdaMatcher-LoFTR (top row) is compared to LoFTR (bottom row) in MegaDepth datasets. Matches with epipolar error beyond 1×10^{-4} are shown in green lines, and the rest are shown in red. Under scale or viewpoint variations, AdaMatcher-LoFTR performs far superior to LoFTR.

Methods	Scale [1,2)			Scale [2,3)			Scale [3,4)			Scale [4,inf)		
	AUC@5° /AUC@10° /AUC@20°											
SIFT [19]+HardNet [21]	21.19	33.01	45.43	10.77	18.55	28.64	4.64	9.31	16.21	1.86	4.36	8.76
KeyNet [2]+HardNet [21]	34.84	49.08	61.30	23.78	35.88	47.69	10.91	19.39	29.97	5.32	10.51	18.48
Disk [32]	33.68	49.76	63.31	5.5	8.45	11.64	0.24	0.47	0.78	0.09	0.19	0.35
R2D2(MS) [23]	37.84	55.90	70.66	22.67	36.93	51.88	6.63	13.02	22.01	2.13	4.02	7.18
ASLFeat [20]	33.80	50.33	65.12	21.87	35.41	49.68	8.53	16.01	26.50	2.95	6.32	11.84
Disk [32]+SG [28]	45.31	63.04	76.49	32.69	48.86	63.76	12.38	22.47	35.68	3.18	6.97	12.05
SP [8]+SG [28]	50.43	67.64	79.97	39.41	57.78	72.34	19.72	35.22	51.97	10.09	19.62	33.88
SP [8]+SG [28]+Ada	53.56	70.01	81.90	42.32	59.51	73.77	23.77	39.55	56.08	12.63	23.68	37.59
SP [8]+SG [28]+OETR [7]	51.96	68.51	79.95	39.92	56.70	71.34	25.37	41.26	57.78	15.36	28.45	44.27
PDC-Net(H) [31]	51.16	67.72	79.58	40.35	56.71	69.49	16.64	26.72	36.77	4.28	8.14	12.39
LoFTR [29]	60.15	74.68	84.45	49.69	65.72	77.94	24.86	39.67	55.08	10.16	18.74	29.97
AdaMatcher-LoFTR	60.50	74.91	84.30	54.53	70.02	81.17	35.13	50.75	64.87	20.14	33.18	47.41
ASpanFormer [6]	60.92	75.29	85.01	54.60	70.21	81.19	33.41	51.16	66.88	18.03	30.50	44.63
AdaMatcher-ASpan	61.29	75.65	85.41	55.35	71.21	82.10	36.05	53.21	67.87	22.92	35.64	50.40
QuadTree [30]	62.06	**76.19**	**85.91**	53.67	69.83	81.59	31.62	48.54	64.60	14.77	26.17	39.89
AdaMatcher-Quad	**62.42**	76.03	85.42	**56.98**	**71.75**	**82.60**	**41.00**	**58.67**	**73.42**	**26.56**	**42.05**	**56.71**

Table 2. **Evaluation on MegaDepth.** Performance gain from AdaMatcher becomes more prominent when scaling variation between image pairs increases. Also, our proposed method can significantly improve the performance of LoFTR and its variants.

resolution images to inference features and ASLFeat [20] extracts features from multi-scale score maps, these methods cannot explicitly model relative scale ratio between images like AdaMatcher. LoFTR [29] and its variants [6, 30] are trained using ground-truth matches obtained by one-to-one assignment, resulting in the inability to learn the geometry consistency of feature matching. When the scale differences between image pairs are large, the number of matches obtained by one-to-one assignment decreases, which would affect the accuracy of camera pose estimation. Since AdaMatcher eliminates the ambiguity of matching during training, it can achieve significant improvement when inferring image pairs with large-scale variations. It can be seen that our proposed method achieves significant performance gains when applied on LoFTR [29], ASpanFormer [6] and QuadTree Attention [30].

Refinement Module. As mentioned before, adaptive assignment and sub-pixel refinement module could be treated as a refinement network with different extractors and matchers as Patch2Pix [37]. Different from SuperGlue's mutual nearest neighbors constraint, we calculate row matches and column matches separately to get many-to-one and one-to-many matches, and then, refine the sub-pixel position in the descriptor feature map. As shown in Tab.2, after adding Ada as a refinement network for SP [8]+SG [28], we observe a noticeable improvement in the AUC metric.

4.3. Visual Localization

Datasets. In HPatches and MegaDepth we only recover relative pose from feature matches. For real-world applications such as AR navigation or autonomous driving, visual localization with absolute pose estimation is a critical geometrical task. Aachen Day-Night v1.1 dataset [36] is chosen to demonstrate the visual localization ability.

Experimental setup. We use open-sourced hierarchical localization pipeline HLoc proposed in [27] to evaluate on day-night query images. To build feature tracks for detector-free methods, we merge keypoints that are close

Methods	Day (0.25m, 2°)	Day (0.5m, 5°)	Day (1.0m, 10°)	Night (0.25m, 2°)	Night (0.5m, 5°)	Night (1.0m, 10°)
SP [8]+SG [28]	**89.8**	**96.1**	**99.4**	77.0	90.6	**100.0**
SP [8]+SG [28]+Patch2Pix [37]	89.3	95.8	99.2	78.0	90.6	99.0
Patch2Pix [37]	86.4	93.0	97.5	72.3	88.5	97.9
LoFTR-DS [29]	-	-	-	72.8	88.5	99.0
LoFTR-OT [29]	88.7	95.6	99.0	78.5	90.6	99.0
ASpanFormer [6]	89.4	95.6	99.0	77.5	91.6	99.5
AdaMatcher-LoFTR	89.2	96.0	99.3	**79.1**	90.6	99.5
AdaMatcher-Quad	89.2	95.9	99.2	**79.1**	**92.1**	99.5

Table 3. **Visual localization evaluation on the Aachen Day-Night benchmark v1.1.**

to each other (with distance less than 4 pixels) by taking their average location, following [37]. It may not be a perfect solution with degraded sub-pixel level accuracy, but it should be a reasonable way to evaluate AdaMatcher.

Results. As shown in Tab. 3, AdaMatcher outperforms all the other detector-free methods. This should be attributed to the fact that adaptive assignment eliminates geometric inconsistency during training and testing. The performance of the detector-free methods is slightly lower than that of SP [8] + SG [28] on the day queries, probably due to the fact that the detector-free methods require quantification of the matches during the database reconstruction process. On the other hand, for night queries the lighting conditions are darker, making the matching process more difficult. However, with the use of adaptive assignment, the geometric consistency is increased, and the descriptive ability is improved. The improvement in matching ability compensates for the loss of quantification during the mapping process, resulting in higher performance indicators.

4.4. Ablation Study

Sub Modules			Pose Estimation AUC						Precision	
CFI	AA	Refine	@5°	Δ	@10°	Δ	@20°	Δ	@1e-4	Δ
			36.22	-	49.70	-	61.86	-	77.61	-
✓			37.13	+2.5%	51.17	+3.0%	63.56	+2.7%	79.13	+2.0%
✓		✓	38.23	+5.5%	52.16	+4.9%	64.22	+3.8%	78.24	+0.8%
	✓		39.98	+10.4%	54.25	+9.2%	66.52	+7.5%	84.20	+8.5%
✓	✓		40.06	+10.6%	54.39	+9.4%	66.73	+7.9%	84.39	+8.7%
✓	✓	✓	**42.54**	**+17.4%**	**57.18**	**+15.1%**	**69.40**	**+12.2%**	**84.99**	**+9.5%**

Table 4. **Ablation of AdaMatcher.** AdaMatcher recovers more accurate relative pose compared to baseline method LoFTR and all parts are useful modules that bring noticeable performance gain for AdaMatcher.

To fully understand different modules in AdaMatcher and evaluate different design choices, we repeat outdoor experiments on MegaDepth with scale ranges in $[1, +\infty)$, as shown in Tab.4. The first row is the result of our baseline method LoFTR [29], 'CFI' represents the LoFTR module (four sets of self-cross attention layers) is replaced by our Co-visible Feature Interaction (Section 3.1), 'AA' denotes replacing LoFTR's coarse-level matching with our adaptive assignment (Section 3.2), and 'Refine' denotes replacing LoFTR's fine-level matching with our sub-pixel refinement module (Section 3.3). We also report match precision in normalized camera coordinates, with epipolar distance threshold of $1e^{-4}$ [8, 9, 28]. By using CFI module, we can get more accurate matches. When the adaptive assignment is allowed in patch-level matching, the accuracy of relative pose estimation and the precision of matching are greatly improved, which means that adaptive assignment plays a vital role here. And the performance can be further enhanced by adding sub-pixel refinement module.

4.5. Runtime Evaluation

To test the timing of inference, we repeated the outdoor experiments on the Megadepth test set with 4000 image pairs, limiting the maximum number of matches to 1024, and the input images are resized to their longer dimensions equal to 640. As shown in Tab.5, since Adamatcher can get more high-quality matches, its inference speed is slightly slower than LoFTR and SP+SG, but the overall execution time (matching + RANSAC) is reduced due to the improvement of inlier ratio and the matching accuracy.

Runtime	**Adamatcher-LoFTR**				**LoFTR**	**PDCNet**	**SP+SG**
(ms/pair)	CFI	AA	Refine	All			
Matching	23.1	17.3	71.6	157.0	104.9	577.4	86.2
+RANSAC	-	-	-	321.4	324.0	776.9	347.1

Table 5. **Inference time.**

5. Conclusions

In this paper, we find the conventional mutual nearest neighbour standard should bottleneck final performance during patch-level or pixel-level matching. The proposed AdaMatcher allows for adaptive assignment during patch-level matching, which overcomes the ambiguous underlying ground-truth label assignments, and enable estimation of the scale ratio between given image pair. We observe a noticeable performance boost, especially when the scale or viewpoint between image pairs varies. We couple co-visible feature decoding and feature interaction, enabling an additional module to be used later to obtain co-visible area. Particularly, by plugging a dedicated sub-pixel refinement module, we can effectively achieve scale alignment and accurate sub-pixel position regression. We have conducted extensive experiments to study the effect of our findings and demonstrated the superiority of our proposed AdaMatcher. We believe that AdaMatcher will bring new insights to the feature matching community.

References

[1] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, pages 5173–5182, 2017. 6

[2] Axel Barroso-Laguna, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Key. net: Keypoint detection by handcrafted and learned cnn filters. In *ICCV*, pages 5836–5844, 2019. 2, 6, 7

[3] Axel Barroso-Laguna, Yurun Tian, and Krystian Mikolajczyk. Scalenet: A shallow architecture for scale estimation. In *CVPR*, pages 12808–12818, 2022. 2

[4] Axel Barroso-Laguna, Yannick Verdie, Benjamin Busam, and Krystian Mikolajczyk. Hdd-net: Hybrid detector descriptor with mutual interactive learning. In *ACCV*, 2020. 2

[5] Gabriele Berton, Carlo Masone, Valerio Paolicelli, and Barbara Caputo. Viewpoint invariant dense matching for visual geolocalization. In *ICCV*, pages 12169–12178, 2021. 2

[6] Hongkai Chen, Zixin Luo, Lei Zhou, Yurun Tian, Mingmin Zhen, Tian Fang, David Mckinnon, Yanghai Tsin, and Long Quan. Aspanformer: Detector-free image matching with adaptive span transformer. In *ECCV*, pages 20–36, 2022. 2, 3, 4, 6, 7, 8

[7] Ying Chen, Dihe Huang, Shang Xu, Jianlin Liu, and Yong Liu. Guide local feature matching by overlap estimation. In *AAAI*, 2022. 2, 3, 6, 7

[8] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPR*, pages 224–236, 2018. 1, 2, 6, 7, 8

[9] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint description and detection of local features. In *CVPR*, pages 8092–8101, 2019. 8

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6

[11] Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. Cotr: Correspondence transformer for matching across images. In *ICCV*, pages 6207–6217, 2021. 2

[12] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *ICML*, pages 5156–5165, 2020. 6

[13] Xinghui Li, Kai Han, Shuda Li, and Victor Prisacariu. Dual-resolution correspondence networks. *NeurIPS*, 33:17346–17357, 2020. 2, 3

[14] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, pages 2041–2050, 2018. 6

[15] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 6

[16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 5

[17] Tony Lindeberg. Feature detection with automatic scale selection. *IJCV*, 30(2):79–116, 1998. 2

[18] Dongfang Liu, Yiming Cui, Liqi Yan, Christos Mousas, Baijian Yang, and Yingjie Chen. Densernet: Weakly supervised visual localization using multi-scale feature aggregation. In *AAAI*, pages 6101–6109, 2021. 2

[19] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1, 2, 6, 7

[20] Zixin Luo, Lei Zhou, Xuyang Bai, Hongkai Chen, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. Aslfeat: Learning local features of accurate shape and localization. In *CVPR*, pages 6589–6598, 2020. 1, 2, 6, 7

[21] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. *NeurIPS*, 30, 2017. 6, 7

[22] Rémi Pautrat, Viktor Larsson, Martin R Oswald, and Marc Pollefeys. Online invariance selection for local feature descriptors. In *ECCV*, pages 707–724, 2020. 2

[23] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. *NeurIPS*, 32, 2019. 1, 2, 6, 7

[24] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Efficient neighbourhood consensus networks via submanifold sparse convolutions. In *ECCV*, pages 605–621, 2020. 2, 3

[25] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. *NeurIPS*, 31, 2018. 2, 3

[26] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, pages 2564–2571. Ieee, 2011. 1, 2

[27] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, pages 12716–12725, 2019. 7

[28] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, pages 4938–4947, 2020. 2, 6, 7, 8

[29] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, pages 8922–8931, 2021. 2, 3, 4, 5, 6, 7, 8

[30] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. In *ICLR*, 2022. 2, 3, 4, 6, 7

[31] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning accurate dense correspondences and when to trust them. In *CVPR*, pages 5714–5724, 2021. 2, 3, 6, 7

[32] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. *NeurIPS*, 33:14254–14265, 2020. 1, 6, 7

[33] Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning feature descriptors using camera pose supervision. In *ECCV*, pages 757–774, 2020. 6

[34] Olivia Wiles, Sebastien Ehrhardt, and Andrew Zisserman. Co-attention for conditioned image matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15920–15929, 2021. 2

[35] Mubariz Zaffar, Sourav Garg, Michael Milford, Julian Kooij, David Flynn, Klaus McDonald-Maier, and Shoaib Ehsan. Vpr-bench: An open-source visual place recognition evaluation framework with quantifiable viewpoint and appearance change. *IJCV*, 129(7):2136–2174, 2021. 2

[36] Zichao Zhang, Torsten Sattler, and Davide Scaramuzza. Reference pose generation for long-term visual localization via learned features and view synthesis. *IJCV*, 129(4):821–844, 2021. 7

[37] Qunjie Zhou, Torsten Sattler, and Laura Leal-Taixe. Patch2pix: Epipolar-guided pixel-level correspondences. In *CVPR*, pages 4669–4678, 2021. 6, 7, 8