LoFTR: Detector-Free Local Feature Matching with Transformers

Jiaming Sun[1,2*] Zehong Shen[1*] Yuang Wang[1*] Hujun Bao[1] Xiaowei Zhou[1†]
[1]Zhejiang University [2]SenseTime Research

Abstract

We present a novel method for local image feature matching. Instead of performing image feature detection, description, and matching sequentially, we propose to first establish pixel-wise dense matches at a coarse level and later refine the good matches at a fine level. In contrast to dense methods that use a cost volume to search correspondences, we use self and cross attention layers in Transformer to obtain feature descriptors that are conditioned on both images. The global receptive field provided by Transformer enables our method to produce dense matches in low-texture areas, where feature detectors usually struggle to produce repeatable interest points. The experiments on indoor and outdoor datasets show that LoFTR outperforms state-of-the-art methods by a large margin. LoFTR also ranks first on two public benchmarks of visual localization among the published methods. Code is available at our project page: https://zju3dv.github.io/loftr/.

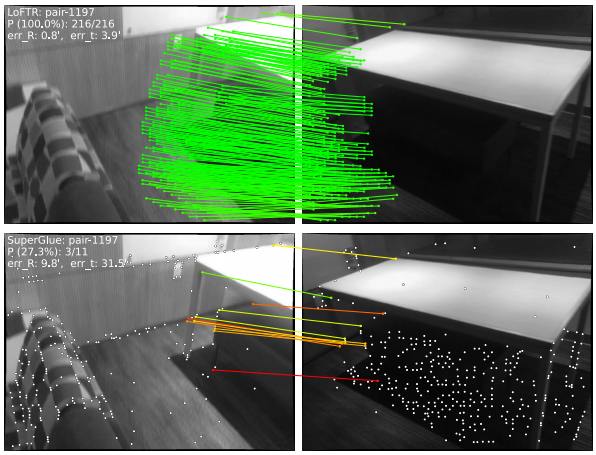

Figure 1: **Comparison between the proposed method LoFTR and the detector-based method SuperGlue [37].** This example demonstrates that LoFTR is capable of finding correspondences on the texture-less wall and the floor with repetitive patterns, where detector-based methods struggle to find repeatable interest points.[1]

1. Introduction

Local feature matching between images is the cornerstone of many 3D computer vision tasks, including structure from motion (SfM), simultaneous localization and mapping (SLAM), visual localization, etc. Given two images to be matched, most existing matching methods consist of three separate phases: feature detection, feature description, and feature matching. In the detection phase, salient points like corners are first detected as interest points from each image. Local descriptors are then extracted around neighborhood regions of these interest points. The feature detection and description phases produce two sets of interest points with descriptors, the point-to-point correspondences of which are later found by nearest neighbor search or more sophisticated matching algorithms.

The use of a feature detector reduces the search space of matching, and the resulted sparse correspondences are sufficient for most tasks, e.g., camera pose estimation. However, a feature detector may fail to extract enough interest points that are repeatable between images due to various factors such as poor texture, repetitive patterns, viewpoint change, illumination variation, and motion blur. This issue is especially prominent in indoor environments, where low-texture regions or repetitive patterns sometimes occupy most areas in the field of view. Fig. 1 shows an example. Without repeatable interest points, it is impossible to find correct correspondences even with perfect descriptors.

Several recent works [34, 33, 19] have attempted to remedy this problem by establishing pixel-wise dense matches. Matches with high confidence scores can be selected from the dense matches, and thus feature detection is avoided. However, the dense features extracted by convolutional neural networks (CNNs) in these works have limited receptive field which may not distinguish indistinctive regions. Instead, humans find correspondences in these indistinctive regions not only based on the *local* neighborhood, but with a larger *global* context. For example, low-texture regions in

*The first three authors contributed equally. The authors are affiliated with the State Key Lab of CAD&CG and ZJU-SenseTime Joint Lab of 3D Vision. †Corresponding author: Xiaowei Zhou.

[1]Only the inlier matches after RANSAC are shown. The green color indicates a match with epipolar error smaller than 5×10^{-4} (in the normalized image coordinates).

Fig. 1 can be distinguished according to their relative positions to the edges. This observation tells us that a large *receptive field* in the feature extraction network is crucial.

Motivated by the above observations, we propose Local Feature TRansformer (LoFTR), a novel detector-free approach to local feature matching. Inspired by seminal work SuperGlue [37], we use Transformer [48] with self and cross attention layers to process (transform) the dense local features extracted from the convolutional backbone. Dense matches are first extracted between the two sets of transformed features at a low feature resolution ($1/8$ of the image dimension). Matches with high confidence are selected from these dense matches and later refined to a sub-pixel level with a correlation-based approach. The global receptive field and positional encoding of Transformer enable the transformed feature representations to be context- and position-dependent. By interleaving the self and cross attention layers multiple times, LoFTR learns the densely-arranged globally-consented matching priors exhibited in the ground-truth matches. A linear transformer is also adopted to reduce the computational complexity to a manageable level.

We evaluate the proposed method on several image matching and camera pose estimation tasks with indoor and outdoor datasets. The experiments show that LoFTR outperforms detector-based and detector-free feature matching baselines by a large margin. LoFTR also achieves state-of-the-art performance and ranks first among the published methods on two public benchmarks of visual localization. Compared to detector-based baseline methods, LoFTR can produce high-quality matches even in indistinctive regions with low-textures, motion blur, or repetitive patterns.

2. Related Work

Detector-based Local Feature Matching. Detector-based methods have been the dominant approach for local feature matching. Before the age of deep learning, many renowned works in the traditional hand-crafted local features have achieved good performances. SIFT [26] and ORB [35] are arguably the most successful hand-crafted local features and are widely adopted in many 3D computer vision tasks. The performance on large viewpoint and illumination changes of local features can be significantly improved with learning-based methods. Notably, LIFT [51] and MagicPoint [8] are among the first successful learning-based local features. They adopt the detector-based design in hand-crafted methods and achieve good performance. SuperPoint [9] builds upon MagicPoint and proposes a self-supervised training method through homographic adaptation. Many learning-based local features along this line [32, 11, 25, 28, 47] also adopt the detector-based design.

The above-mentioned local features use the nearest neighbor search to find matches between the extracted interest points. Recently, SuperGlue [37] proposes a learning-based approach for local feature matching. SuperGlue accepts two sets of interest points with their descriptors as input and learns their matches with a graph neural network (GNN), which is a general form of Transformers [16]. Since the priors in feature matching can be learned with a data-driven approach, SuperGlue achieves impressive performance and sets the new state of the art in local feature matching. However, being a detector-dependent method, it has the fundamental drawback of being unable to detect repeatable interest points in indistinctive regions. The attention range in SuperGlue is also limited to the detected interest points only. Our work is inspired by SuperGlue in terms of using self and cross attention in GNN for message passing between two sets of descriptors, but we propose a detector-free design to avoid the drawbacks of feature detectors. We also use an efficient variant of the attention layers in Transformer to reduce the computation costs.

Detector-free Local Feature Matching. Detector-free methods remove the feature detector phase and directly produce dense descriptors or dense feature matches. The idea of dense features matching dates back to SIFT Flow [23]. [6, 39] are the first learning-based approaches to learn pixel-wise feature descriptors with the contrastive loss. Similar to the detector-based methods, the nearest neighbor search is usually used as a post-processing step to match the dense descriptors. NCNet [34] proposed a different approach by directly learning the dense correspondences in an end-to-end manner. It constructs 4D cost volumes to enumerate all the possible matches between the images and uses 4D convolutions to regularize the cost volume and enforce neighborhood consensus among all the matches. Sparse NCNet [33] improves upon NCNet and makes it more efficient with sparse convolutions. Concurrently with our work, DRC-Net [19] follows this line of work and proposes a coarse-to-fine approach to produce dense matches with higher accuracy. Although all the possible matches are considered in the 4D cost volume, the receptive field of 4D convolution is still limited to each matches' neighborhood area. Apart from neighborhood consensus, our work focuses on achieving global consensus between matches with the help of the global receptive field in Transformers, which is not exploited in NCNet and its follow-up works. [24] proposes a dense matching pipeline for SfM with endoscopy videos. The recent line of research [46, 45, 44, 15] that focuses on bridging the task of local feature matching and optical flow estimation, is also related to our work.

Transformers in Vision Related Tasks. Transformer [48] has become the *de facto* standard for sequence modeling in natural language processing (NLP) due to their simplic-

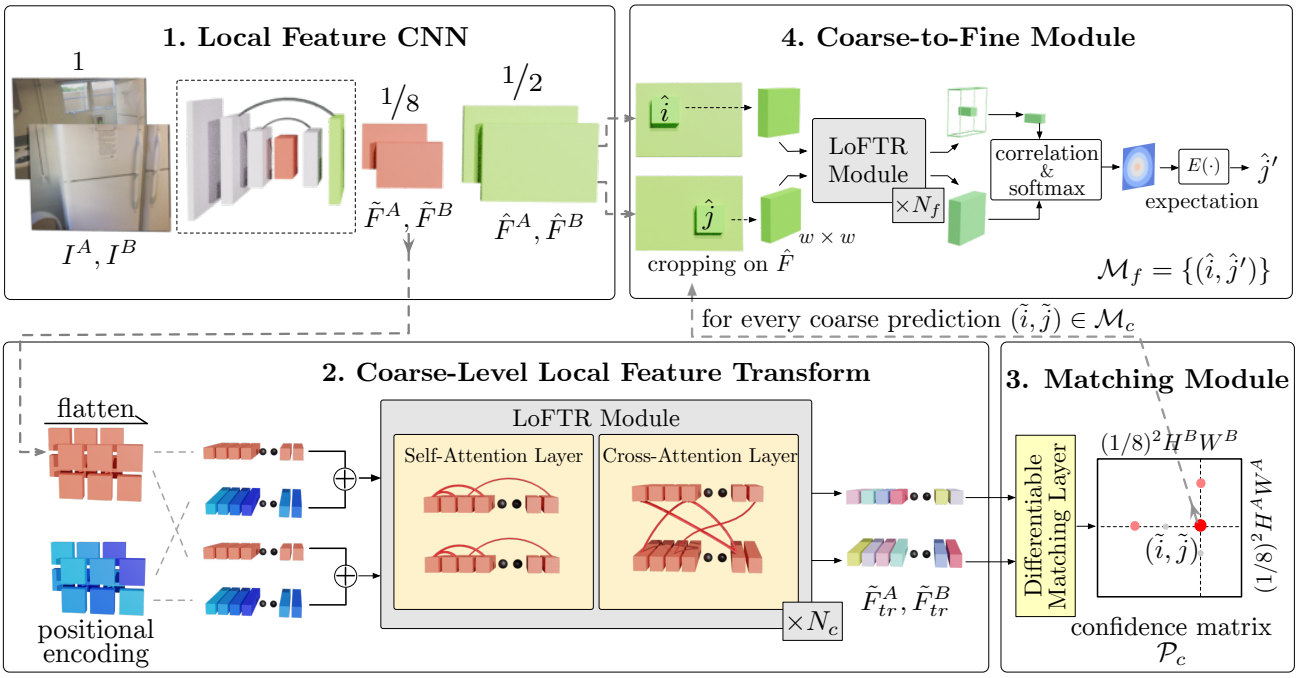

Figure 2: **Overview of the proposed method.** LoFTR has four components: **1.** A local feature CNN extracts the coarse-level feature maps $\tilde{F}^A$ and $\tilde{F}^B$, together with the fine-level feature maps $\hat{F}^A$ and $\hat{F}^B$ from the image pair I^A and I^B (Section 3.1). **2.** The coarse feature maps are flattened to 1-D vectors and added with the positional encoding. The added features are then processed by the Local Feature TRansformer (LoFTR) module, which has N_c self-attention and cross-attention layers (Section 3.2). **3.** A differentiable matching layer is used to match the transformed features, which ends up with a confidence matrix $\mathcal{P}_c$. The matches in $\mathcal{P}_c$ are selected according to the confidence threshold and mutual-nearest-neighbor criteria, yielding the coarse-level match prediction $\mathcal{M}_c$ (Section 3.3). **4.** For every selected coarse prediction $(\tilde{i}, \tilde{j}) \in \mathcal{M}_c$, a local window with size $w \times w$ is cropped from the fine-level feature map. Coarse matches will be refined within this local window to a sub-pixel level as the final match prediction $\mathcal{M}_f$ (Section 3.4).

ity and computation efficiency. Recently, Transformers are also getting more attention in computer vision tasks, such as image classification [10], object detection [3] and semantic segmentation [49]. Concurrently with our work, [20] proposes to use Transformer for disparity estimation. The computation cost of the vanilla Transformer grows quadratically as the length of input sequences due to the multiplication between query and key vectors. Many efficient variants [42, 18, 17, 5] are proposed recently in the context of processing long language sequences. Since no assumption of the input data is made in these works, they are also well suited for processing images.

3. Methods

Given the image pair I^A and I^B, the existing local feature matching methods use a feature detector to extract interest points. We propose to tackle the repeatability issue of feature detectors with a detector-free design. An overview of the proposed method LoFTR is presented in Fig. 2.

3.1. Local Feature Extraction

We use a standard convolutional architecture with FPN [22] (denoted as the local feature CNN) to extract multi-level features from both images. We use $\tilde{F}^A$ and $\tilde{F}^B$ to denote the coarse-level features at $1/8$ of the original image dimension, and $\hat{F}^A$ and $\hat{F}^B$ the fine-level features at $1/2$ of the original image dimension.

Convolutional Neural Networks (CNNs) possess the inductive bias of translation equivariance and locality, which are well suited for *local* feature extraction. The downsampling introduced by the CNN also reduces the input length of the LoFTR module, which is crucial to ensure a manageable computation cost.

3.2. Local Feature Transformer (LoFTR) Module

After the local feature extraction, $\tilde{F}^A$ and $\tilde{F}^B$ are passed through the LoFTR module to extract position and context dependent local features. Intuitively, the LoFTR module transforms the features into feature representations that are easy to match. We denote the transformed features as $\tilde{F}^A_{tr}$ and $\tilde{F}^B_{tr}$.

Preliminaries: Transformer [48]. We first briefly introduce the Transformer here as background. A Transformer encoder is composed of sequentially connected encoder layers. Fig. 3(a) shows the architecture of an encoder layer.

The key element in the encoder layer is the attention

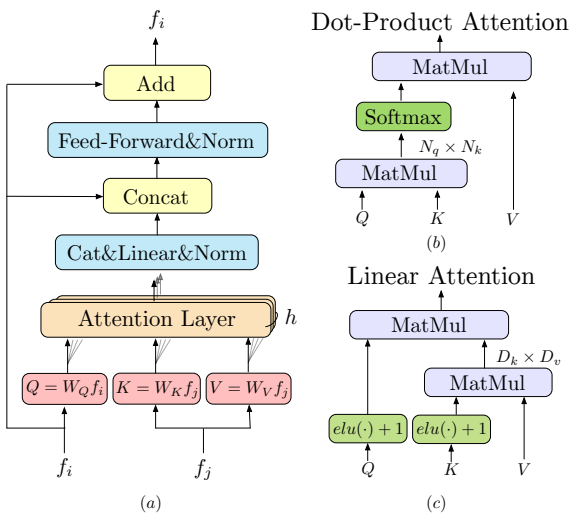

Figure 3: **Encoder layer and attention layer in LoFTR.** (a) Transformer encoder layer. h represents the multiple heads of attention. (b) Vanilla dot-product attention with $O(N^2)$ complexity. (c) Linear attention layer with $O(N)$ complexity. The scale factor is omitted for simplicity.

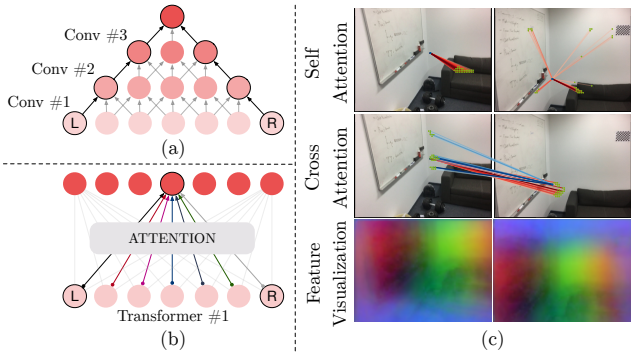

Figure 4: Illustration of the receptive field of **(a)** Convolutions and **(b)** Transformers. Assume that the objective is to establish a connection between the L and R elements to extract their joint feature representation. Due to the local-connectivity of convolutions, many convolution layers need to be stacked together in order to achieve this connection. The global receptive field of Transformers enables this connection to be established through only one attention layer. **(c)** Visualization of the attention weights and transformed dense features. We use PCA to reduce the dimension of the transformed features $\tilde{F}^A_{tr}$ and $\tilde{F}^B_{tr}$ and visualize the results with RGB color. Zoom in for details.

layer. The input vectors for an attention layer are conventionally named query, key, and value. Analogous to information retrieval, the query vector Q retrieves information from the value vector V, according to the attention weight computed from the dot product of Q and the key vector K corresponding to each value V. The computation graph of the attention layer is presented in Fig. 3(b). Formally, the attention layer is denoted as:

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T)V.$$

Intuitively, the attention operation selects the relevant information by measuring the similarity between the query element and each key element. The output vector is the sum of the value vectors weighted by the similarity scores. As a result, the relevant information is extracted from the value vector if the similarity is high. This process is also called "message passing" in Graph Neural Network.

Linear Transformer. Denoting the length of Q and K as N and their feature dimension as D, the dot product between Q and K in the Transformer introduces computation cost that grows quadratically ($O(N^2)$) with the length of the input sequence. Directly applying the vanilla version of Transformer in the context of local feature matching is impractical even when the input length is reduced by the local feature CNN. To remedy this problem, we propose to use an efficient variant of the vanilla attention layer in Transformer. Linear Transformer [17] proposes to reduce the computation complexity of Transformer to $O(N)$ by substituting the exponential kernel used in the original attention layer with an alternative kernel function $\text{sim}(Q, K) = \phi(Q) \cdot \phi(K)^T$, where $\phi(\cdot) = \text{elu}(\cdot) + 1$. This operation is illustrated by the computation graph in Fig. 3(c). Utilizing the associativity property of matrix products, the multiplication between $\phi(K)^T$ and V can be carried out first. Since $D \ll N$, the computation cost is reduced to $O(N)$.

Positional Encoding. We use the 2D extension of the standard positional encoding in Transformers following DETR [3]. Different from DETR, we only add them to the backbone output once. We leave the formal definition of the positional encoding in the supplementary material. Intuitively, the positional encoding gives each element unique position information in the sinusoidal format. By adding the position encoding to $\tilde{F}^A$ and $\tilde{F}^B$, the transformed features will become position-dependent, which is crucial to the ability of LoFTR to produce matches in indistinctive regions. As shown in the bottom row of Fig. 4(c), although the input RGB color is homogeneous on the white walls, the transformed features $\tilde{F}^A_{tr}$ and $\tilde{F}^B_{tr}$ are *unique* for each position demonstrated by the smooth color gradients. More visualizations are provided in Fig. 6.

Self-attention and Cross-attention Layers. For self-attention layers, the input features f_i and f_j (shown in Fig. 3) are the same (either $\tilde{F}^A$ or $\tilde{F}^B$). For cross-attention layers, the input features f_i and f_j are either ($\tilde{F}^A$ and $\tilde{F}^B$) or ($\tilde{F}^B$ and $\tilde{F}^A$) depending on the direction of cross-attention. Following [37], we interleave the self and cross attention layers in the LoFTR module by N_c times. The attention weights of the self and cross attention layers in LoFTR are visualized in the first two rows of Fig. 4(c).

3.3. Establishing Coarse-level Matches

Two types of differentiable matching layers can be applied in LoFTR, either with an optimal transport (OT) layer as in [37] or with a dual-softmax operator [34, 47]. The score matrix $\mathcal{S}$ between the transformed features is first calculated by $\mathcal{S}(i,j) = \frac{1}{\tau} \cdot \langle \tilde{F}^A_{tr}(i), \tilde{F}^B_{tr}(j) \rangle$. When matching with OT, $-\mathcal{S}$ can be used as the cost matrix of the partial assignment problem as in [37]. We can also apply softmax on both dimensions (referred to as dual-softmax in the following) of $\mathcal{S}$ to obtain the probability of soft mutual nearest neighbor matching. Formally, when using dual-softmax, the matching probability $\mathcal{P}_c$ is obtained by:

$$\mathcal{P}_c(i,j) = \text{softmax}\left(\mathcal{S}(i,\cdot)\right)_j \cdot \text{softmax}\left(\mathcal{S}(\cdot,j)\right)_i .$$

Match Selection. Based on the confidence matrix $\mathcal{P}_c$, we select matches with confidence higher than a threshold of θ_c, and further enforce the mutual nearest neighbor (MNN) criteria, which filters possible outlier coarse matches. We denote the coarse-level match predictions as:

$$\mathcal{M}_c = \{(\tilde{i},\tilde{j}) \mid \forall (\tilde{i},\tilde{j}) \in \text{MNN}(\mathcal{P}_c),\ \mathcal{P}_c(\tilde{i},\tilde{j}) \geq \theta_c\}.$$

3.4. Coarse-to-Fine Module

After establishing coarse matches, these matches are refined to the original image resolution with the coarse-to-fine module. Inspired by [50], we use a correlation-based approach for this purpose. For every coarse match $(\tilde{i},\tilde{j})$, we first locate its position $(\hat{i},\hat{j})$ at fine-level feature maps $\hat{F}^A$ and $\hat{F}^B$, and then crop two sets of local windows of size $w \times w$. A smaller LoFTR module then transforms the cropped features within each window by N_f times, yielding two transformed local feature maps $\hat{F}^A_{tr}(\hat{i})$ and $\hat{F}^B_{tr}(\hat{j})$ centered at $\hat{i}$ and $\hat{j}$, respectively. Then, we correlate the center vector of $\hat{F}^A_{tr}(\hat{i})$ with all vectors in $\hat{F}^B_{tr}(\hat{j})$ and thus produce a heatmap that represents the matching probability of each pixel in the neighborhood of $\hat{j}$ with $\hat{i}$. By computing expectation over the probability distribution, we get the final position $\hat{j}'$ with sub-pixel accuracy on I^B. Gathering all the matches $\{(\hat{i},\hat{j}')\}$ produces the final fine-level matches $\mathcal{M}_f$.

3.5. Supervision

The final loss consists of the losses for the coarse-level and the fine-level: $\mathcal{L} = \mathcal{L}_c + \mathcal{L}_f$.

Coarse-level Supervision. The loss function for the coarse-level is the negative log-likelihood loss over the confidence matrix $\mathcal{P}_c$ returned by either the optimal transport layer or the dual-softmax operator. We follow SuperGlue [37] to use camera poses and depth maps to compute the ground-truth labels for the confidence matrix during training. We define the ground-truth coarse matches $\mathcal{M}^{gt}_c$ as the mutual nearest neighbors of the two sets of 1/8-resolution grids. The distance between two grids is measured by the re-projection distance of their central locations. More details are provided in the supplementary. With the optimal transport layer, we use the same loss formulation as in [37]. When using dual-softmax for matching, we minimize the negative log-likelihood loss over the grids in $\mathcal{M}^{gt}_c$:

$$\mathcal{L}_c = -\frac{1}{|\mathcal{M}^{gt}_c|} \sum_{(\tilde{i},\tilde{j}) \in \mathcal{M}^{gt}_c} \log \mathcal{P}_c\left(\tilde{i},\tilde{j}\right).$$

Fine-level Supervision. We use the ℓ_2 loss for fine-level refinement. Following [50], for each query point $\hat{i}$, we also measure its uncertainty by calculating the total variance $\sigma^2(\hat{i})$ of the corresponding heatmap. The target is to optimize the refined position that has low uncertainty, resulting in the final weighted loss function:

$$\mathcal{L}_f = \frac{1}{|\mathcal{M}_f|} \sum_{(\hat{i},\hat{j}') \in \mathcal{M}_f} \frac{1}{\sigma^2(\hat{i})} \left\| \hat{j}' - \hat{j}'_{gt} \right\|_2 ,$$

in which $\hat{j}'_{gt}$ is calculated by warping each $\hat{i}$ from $\hat{F}^A_{tr}(\hat{i})$ to $\hat{F}^B_{tr}(\hat{j})$ with the ground-truth camera pose and depth. We ignore $(\hat{i},\hat{j}')$ if the warped location of $\hat{i}$ falls out of the local window of $\hat{F}^B_{tr}(\hat{j})$ when calculating $\mathcal{L}_f$. The gradient is not backpropagated through $\sigma^2(\hat{i})$ during training.

3.6. Implementation Details

We train the indoor model of LoFTR on the ScanNet [7] dataset and the outdoor model on the MegaDepth [21] following [37]. On ScanNet, the model is trained using Adam with an initial learning rate of 1×10^{-3} and a batch size of 64. It converges after 24 hours of training on 64 GTX 1080Ti GPUs. The local feature CNN uses a modified version of ResNet-18 [12] as the backbone. The entire model is trained end-to-end with randomly initialized weights. N_c is set to 4 and N_f is 1. θ_c is chosen to 0.2. Window size w is equal to 5. $\tilde{F}^A_{tr}$ and $\tilde{F}^B_{tr}$ are upsampled and concatenated with $\hat{F}^A$ and $\hat{F}^B$ before passing through the fine-level LoFTR in the implementation. The full model with dual-softmax matching runs at 116 ms for a 640×480 image pair on an RTX 2080Ti. Under the optimal transport setup, we use three sinkhorn iterations, and the model runs at 130 ms. We refer readers to the supplementary material for more details of training and timing analyses.

4. Experiments

4.1. Homography Estimation

In the first experiment, we evaluate LoFTR on the widely adopted HPatches dataset [1] for homography estimation. HPatches contains 52 sequences under significant illumination changes and 56 sequences that exhibit large variation in viewpoints.

Category	Method	Homography est. AUC			#matches
		@3px	@5px	@10px	
Detector-based	D2Net [11]+NN	23.2	35.9	53.6	0.2K
	R2D2 [32]+NN	50.6	63.9	76.8	0.5K
	DISK [47]+NN	52.3	64.9	78.9	1.1K
	SP [9]+SuperGlue [37]	53.9	68.3	81.7	0.6K
Detector-free	Sparse-NCNet [33]	48.9	54.2	67.1	1.0K
	DRC-Net [19]	50.6	56.2	68.3	1.0K
	LoFTR-DS	**65.9**	**75.6**	**84.6**	1.0K

Table 1: **Homography estimation on HPatches [7].** The AUC of the corner error in percentage is reported. The suffix DS indicates the differentiable matching with dual-softmax.

Evaluation protocol. In every test sequence, one reference image is paired with the rest five images. All images are resized with shorter dimensions equal to 480. For each image pair, we extract a set of matches with LoFTR trained on MegaDepth [21]. We use OpenCV to compute the homography estimation with RANSAC as the robust estimator. To make a fair comparison to methods that produce different numbers of matches, we compute the corner error between the images warped with the estimated $\hat{\mathcal{H}}$ and the ground-truth $\mathcal{H}$ as a correctness identifier as in [9]. Following [37], we report the area under the cumulative curve (AUC) of the corner error up to threshold values of 3, 5, and 10 pixels, respectively. We report the results of LoFTR with a maximum of 1K output matches.

Baseline methods. We compare LoFTR with three categories of methods: 1) detector-based local features including R2D2 [32], D2Net [11], and DISK [47], 2) a detector-based local feature matcher, i.e., SuperGlue [37] on top of SuperPoint [9] features, and 3) detector-free matchers including Sparse-NCNet [33] and DRC-Net [19]. For local features, we extract a maximum of 2K features with which we extract mutual nearest neighbors as the final matches. For methods directly outputting matches, we restrict a maximum of 1K matches, same as LoFTR. We use the default hyperparameters in the original implementations for all the baselines.

Results. Tab. 1 shows that LoFTR notably outperforms other baselines under all error thresholds by a significant margin. Specifically, the performance gap between LoFTR and other methods increases with a stricter correctness threshold. We attribute the top performance to the larger number of match candidates provided by the detector-free design and the global receptive field brought by the Transformer. Moreover, the coarse-to-fine module also contributes to the estimation accuracy by refining matches to a sub-pixel level.

4.2. Relative Pose Estimation

Datasets. We use ScanNet [7] and MegaDepth [21] to demonstrate the effectiveness of LoFTR for pose estimation in indoor and outdoor scenes, respectively.

Category	Method	Pose estimation AUC		
		@5°	@10°	@20°
Detector-based	ORB [35]+GMS [2]	5.21	13.65	25.36
	D2-Net [11]+NN	5.25	14.53	27.96
	ContextDesc [27]+Ratio Test [26]	6.64	15.01	25.75
	SP [9]+NN	9.43	21.53	36.40
	SP [9]+PointCN [52]	11.40	25.47	41.41
	SP [9]+OANet [53]	11.76	26.90	43.85
	SP [9]+SuperGlue [37]	16.16	33.81	51.84
Detector-free	DRC-Net † [19]	7.69	17.93	30.49
	LoFTR-OT†	16.88	33.62	50.62
	LoFTR-OT	21.51	40.39	**57.96**
	LoFTR-DS	**22.06**	**40.8**	57.62

Table 2: **Evaluation on ScanNet [7] for indoor pose estimation.** The AUC of the pose error in percentage is reported. LoFTR improves the state-of-the-art methods by a large margin. †indicates models trained on MegaDepth. The suffixes OT and DS indicate differentiable matching with optimal transport and dual-softmax, respectively.

Category	Method	Pose estimation AUC		
		@5°	@10°	@20°
Detector-based	SP [9]+SuperGlue [37]	42.18	61.16	75.96
Detector-free	DRC-Net [19]	27.01	42.96	58.31
	LoFTR-OT	50.31	67.14	79.93
	LoFTR-DS	**52.8**	**69.19**	**81.18**

Table 3: **Evaluation on MegaDepth [21] for outdoor pose estimation.** Matching with LoFTR results in better performance in the outdoor pose estimation task.

ScanNet contains 1613 monocular sequences with ground truth poses and depth maps. Following the procedure from SuperGlue [37], we sample 230M image pairs for training, with overlap scores between 0.4 and 0.8. We evaluate our method on the 1500 testing pairs from [37]. All images and depth maps are resized to 640×480. This dataset contains image pairs with wide baselines and extensive texture-less regions.

MegaDepth consists of 1M internet images of 196 different outdoor scenes. The authors also provide sparse reconstruction from COLMAP [40] and depth maps computed from multi-view stereo. We follow DISK [47] to only use the scenes of "Sacre Coeur" and "St. Peter's Square" for validation, from which we sample 1500 pairs for a fair comparison. Images are resized such that their longer dimensions are equal to 840 for training and 1200 for validation. The key challenge on MegaDepth is matching under extreme viewpoint changes and repetitive patterns.

Evaluation protocol. Following [37], we report the AUC of the pose error at thresholds $(5^\circ, 10^\circ, 20^\circ)$, where the pose error is defined as the maximum of angular error in rotation and translation. To recover the camera pose, we solve the essential matrix from predicted matches with RANSAC. We don't compare the matching precisions between LoFTR and other detector-based methods due to the lack of a well-

Method	Day	Night
	(0.25m,2°) / (0.5m,5°) / (1.0m,10°)	
Local Feature Evaluation on Night-time Queries		
R2D2 [32]+NN	-	71.2 / 86.9 / 98.9
LISRD [31]+SP [9]+AdaLam [4]	-	**73.3** / 86.9 / 97.9
ISRF [29]+NN	-	69.1 / 87.4 / 98.4
SP [9]+SuperGlue [37]	-	**73.3** / 88.0 / 98.4
LoFTR-DS	-	72.8 / **88.5** / **99.0**
Full Visual Localization with HLoc		
SP [9]+SuperGlue [37]	**89.8** / **96.1** / **99.4**	77.0 / **90.6** / **100.0**
LoFTR-OT	88.7 / 95.6 / 99.0	**78.5** / **90.6** / 99.0

Table 4: **Visual localization evaluation on the Aachen Day-Night [54] benchmark v1.1**. The evaluation results on both the local feature evaluation track and the full visual localization track are reported.

Method	DUC1	DUC2
	(0.25m,10°) / (0.5m,10°) / (1.0m,10°)	
ISRF [29]	39.4 / 58.1 / 70.2	41.2 / 61.1 / 69.5
KAPTURE [14]+R2D2 [32]	41.4 / 60.1 / 73.7	47.3 / 67.2 / 73.3
HLoc [36]+SP [9]+SuperGlue [37]	**49.0** / 68.7 / 80.8	53.4 / **77.1** / 82.4
HLoc [36]+**LoFTR-OT**	47.5 / **72.2** / **84.8**	**54.2** / 74.8 / **85.5**

Table 5: **Visual localization evaluation on the InLoc [41] benchmark**.

Method	Pose estimation AUC		
	@5°	@10°	@20°
1) replace LoFTR with convolution	14.98	32.04	49.92
2) 1/16 coarse-resolution + 1/4 fine-resolution	16.75	34.82	54.0
3) positional encoding per layer	18.02	35.64	52.77
4) larger model with $N_c = 8$, $N_f = 2$	**20.87**	40.23	57.56
Full ($N_c = 4$, $N_f = 1$)	20.06	**40.8**	**57.62**

Table 6: **Ablation study.** Five variants of LoFTR are trained and evaluated both on the ScanNet dataset.

defined metric (e.g., matching score or recall [13, 30]) for detector-free image matching methods. We consider DRC-Net [19] as the state-of-the-art method in detector-free approaches [34, 33].

Results of indoor pose estimation. LoFTR achieves the best performance in pose accuracy compared to all competitors (see Tab. 2 and Fig. 5). Pairing LoFTR with optimal transport or dual-softmax as the differentiable matching layer achieves comparable performance. Since the released model of DRC-Net† is trained on MegaDepth, we provide the results of LoFTR† trained on MegaDepth for a fair comparison. LoFTR† also outperforms DRC-Net† by a large margin in this evaluation (see Fig. 5), which demonstrates the generalizability of our model across datasets.

Results of Outdoor Pose Estimation. As shown in Tab. 3, LoFTR outperforms the detector-free method DRC-Net by 61% at AUC@10°, demonstrating the effectiveness of the Transformer. For SuperGlue, we use the setup from the open-sourced localization toolbox HLoc [36]. LoFTR outperforms SuperGlue by a large margin (13% at AUC@10°), which demonstrates the effectiveness of the detector-free design. Different from indoor scenes, LoFTR-DS performs better than LoFTR-OT on MegaDepth. More qualitative results can be found in Fig. 5.

4.3. Visual Localization

Visual Localization. Besides achieving competitive performance for relative pose estimation, LoFTR can also benefit visual localization, which is the task to estimate the 6-DoF poses of given images with respect to the corresponding 3D scene model. We evaluate LoFTR on the Long-Term Visual Localization Benchmark [43] (referred to as VisLoc benchmark in the following). It focuses on benchmarking visual localization methods under varying conditions, e.g., day-night changes, scene geometry changes, and indoor scenes with plenty of texture-less areas. Thus, the visual localization task relies on highly robust image matching methods.

Evaluation. We evaluate LoFTR on two tracks of VisLoc that consist of several challenges. First, the "visual localization for handheld devices" track requires a full localization pipeline. It benchmarks on two datasets, the Aachen-Day-Night dataset [38, 54] concerning outdoor scenes and the InLoc [41] dataset concerning indoor scenes. We use open-sourced localization pipeline HLoc [36] with the matches extracted by LoFTR. Second, the "local features for long-term localization" track provides a fixed localization pipeline to evaluate the local feature extractors themselves and optionally the matchers. This track uses the Aachen v1.1 dataset [54]. We provide the implementation details of testing LoFTR on VisLoc in the supplementary material.

Results. We provide evaluation results of LoFTR in Tab. 4 and Tab. 5. We have evaluated LoFTR pairing with either the optimal transport layer or the dual-softmax operator and report the one with better results. LoFTR-DS outperforms all baselines in the local feature challenge track, showing its robustness under day-night changes. Then, for the visual localization for handheld devices track, LoFTR-OT outperforms all published methods on the challenging InLoc dataset, which contains extensive appearance changes, more texture-less areas, symmetric and repetitive elements. We attribute the prominence to the use of the Transformer and the optimal transport layer, taking advantage of global information and jointly bringing global consensus into the final matches. The detector-free design also plays a critical role, preventing the repeatability problem of detector-based methods in low-texture regions. LoFTR-OT performs on par with the state-of-the-art method SuperPoint + SuperGlue on night queries of the Aachen v1.1 dataset and slightly worse on the day queries.

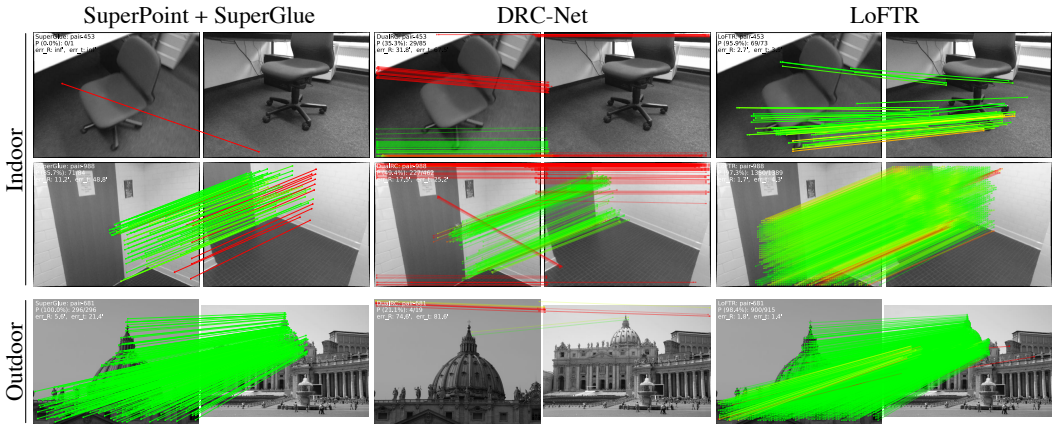

Figure 5: **Qualitative results.** LoFTR is compared to SuperGlue [37] and DRC-Net [19] in indoor and outdoor environments. LoFTR obtains more correct matches and fewer mismatches, successfully coping with low-texture regions and large viewpoint and illumination changes. The red color indicates epipolar error beyond 5×10^{-4} for indoor scenes and 1×10^{-4} for outdoor scenes (in the normalized image coordinates). More qualitative results can be found on the project webpage.

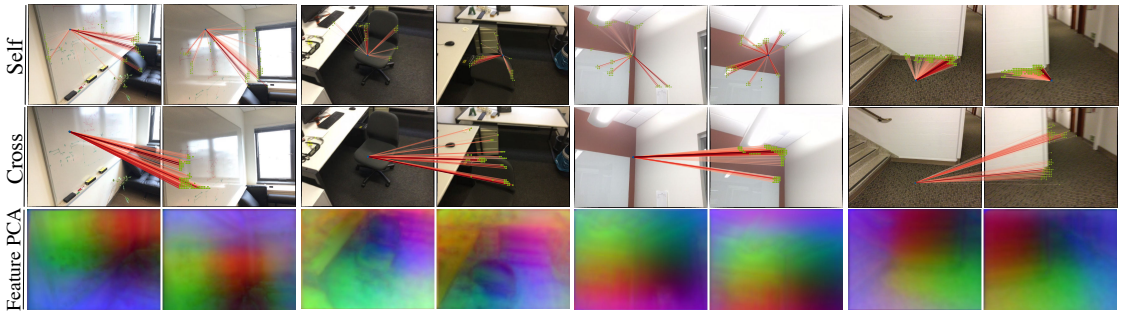

Figure 6: **Visualization of self and cross attention weights and the transformed features.** In the first two examples, the query point from the low-texture region is able to aggregate the surrounding global information flexibly. For instance, the point on the chair is looking at the edge of the chair. In the last two examples, the query point from the distinctive region can also utilize the richer information from other regions. The feature visualization with PCA further shows that LoFTR learns a position-dependent feature representation.

4.4. Understanding LoFTR

Ablation Study. To fully understand the different modules in LoFTR, we evaluate five different variants with results shown in Tab. 6: 1) Replacing the LoFTR module by convolution with a comparable number of parameters results in a significant drop in AUC as expected. 2) Using a smaller version of LoFTR with 1/16 and 1/4 resolution feature maps at the coarse and fine level, respectively, results in a running time of 104 ms and a degraded pose estimation accuracy. 3) Using DETR-style [3] Transformer architecture which has positional encoding at each layer, leads to a noticeably declined result. 4) Increasing the model capacity by doubling the number of LoFTR layers to $N_c = 8$ and $N_f = 2$ barely changes the results. We conduct these experiments using the same training and evaluation protocol as indoor pose estimation on ScanNet with an optimal transport layer for matching.

Visualizing Attention. We visualize the attention weights in Fig. 6.

5. Conclusion

This paper presents a novel detector-free matching approach, named LoFTR, that can establish accurate semi-dense matches with Transformers in a coarse-to-fine manner. The proposed LoFTR module uses the self and cross attention layers in Transformers to transform the local features to be context- and position-dependent, which is crucial for LoFTR to obtain high-quality matches on indistinctive regions with low-texture or repetitive patterns. Our experiments show that LoFTR achieves state-of-the-art performances on relative pose estimation and visual localization on multiple datasets. We believe that LoFTR provides a new direction for detector-free methods in local image feature matching and can be extended to more challenging scenarios, e.g., matching images with severe seasonal changes.

Acknowledgement. The authors would like to acknowledge the support from the National Key Research and Development Program of China (No. 2020AAA0108901), NSFC (No. 61806176), and ZJU-SenseTime Joint Lab of 3D Vision.

References

[1] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, 2017.

[2] JiaWang Bian, Wen-Yan Lin, Yasuyuki Matsushita, Sai-Kit Yeung, Tan-Dat Nguyen, and Ming-Ming Cheng. GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence. In *CVPR*, 2017.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[4] Luca Cavalli, Viktor Larsson, Martin Ralf Oswald, Torsten Sattler, and Marc Pollefeys. Handcrafted Outlier Detection Revisited. In *ECCV*, 2020.

[5] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *ICLR*, 2021.

[6] Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. *NeurIPS*, 2016.

[7] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017.

[8] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Toward geometric deep slam. *arXiv:1707.07410*.

[9] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *CVPRW*, 2018.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[11] 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. *CVPR*, 2019.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[13] Jared Heinly, Enrique Dunn, and Jan-Michael Frahm. Comparative evaluation of binary features. In *ECCV*, 2012.

[14] Martin Humenberger, Yohann Cabon, Nicolas Guerin, Julien Morat, Jérôme Revaud, Philippe Rerole, Noé Pion, Cesar de Souza, Vincent Leroy, and Gabriela Csurka. Robust Image Retrieval-based Visual Localization using Kapture. *arXiv:2007.13867*.

[15] Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. COTR: Correspondence Transformer for Matching Across Images, 2021.

[16] Chaitanya Joshi. Transformers are Graph Neural Networks. https://thegradient.pub/transformers-are-graph-neural-networks/, 2020.

[17] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *ICML*, 2020.

[18] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *ICLR*, 2020.

[19] Xinghui Li, Kai Han, Shuda Li, and Victor Prisacariu. Dual-resolution correspondence networks. *NeurIPS*, 2020.

[20] Zhaoshuo Li, Xingtong Liu, Francis X Creighton, Russell H Taylor, and Mathias Unberath. Revisiting Stereo Depth Estimation From a Sequence-to-Sequence Perspective with Transformers. *arXiv:2011.02910*.

[21] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018.

[22] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature Pyramid Networks for Object Detection. *CVPR*, 2017.

[23] Ce Liu, Jenny Yuen, and Antonio Torralba. SIFT Flow: Dense correspondence across scenes and its applications. *T-PAMI*, 2010.

[24] X. Liu, Y. Zheng, B. Killeen, M. Ishii, G. D. Hager, R. H. Taylor, and M. Unberath. Extremely Dense Point Correspondences Using a Learned Feature Descriptor. In *CVPR*, 2020.

[25] Yuan Liu, Zehong Shen, Zhixuan Lin, Sida Peng, Hujun Bao, and Xiaowei Zhou. GIFT: Learning transformation-invariant dense visual descriptors via group cnns. *NeurIPS*, 2019.

[26] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[27] Zixin Luo, Tianwei Shen, Lei Zhou, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. ContextDesc: Local Descriptor Augmentation with Cross-Modality Context. *CVPR*, 2019.

[28] 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*, 2020.

[29] Iaroslav Melekhov, Gabriel J Brostow, Juho Kannala, and Daniyar Turmukhambetov. Image Stylization for Robust Features. *arXiv:2008.06959*.

[30] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *T-PAMI*, 2005.

[31] Rémi Pautrat, Viktor Larsson, Martin R Oswald, and Marc Pollefeys. Online Invariance Selection for Local Feature Descriptors. In *ECCV*, 2020.

[32] Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noe Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2D2: repeatable and reliable detector and descriptor. *NeurIPS*, 2019.

[33] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Efficient neighbourhood consensus networks via submanifold sparse convolutions. In *ECCV*, 2020.

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

[35] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *ICCV*, 2011.

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

[37] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020.

[38] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image Retrieval for Image-Based Localization Revisited. In *BMVC*, 2012.

[39] Tanner Schmidt, Richard Newcombe, and Dieter Fox. Self-supervised visual descriptor learning for dense correspondence. *RAL*, 2016.

[40] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-Motion revisited. In *CVPR*, 2016.

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

[42] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *arXiv:2009.06732*.

[43] Carl Toft, Will Maddern, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, et al. Long-Term Visual Localization Revisited. *T-PAMI*, 2020.

[44] Prune Truong, Martin Danelljan, L. Gool, and R. Timofte. Learning Accurate Dense Correspondences and When to Trust Them. *ArXiv*, abs/2101.01710, 2021.

[45] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. GOCor: Bringing Globally Optimized Correspondence Volumes into Your Neural Network. In *NeurIPS*, 2020.

[46] Prune Truong, Martin Danelljan, and Radu Timofte. GLU-Net: Global-Local Universal Network for dense flow and correspondences. In *CVPR*, 2020.

[47] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. DISK: Learning local features with policy gradient. *NeurIPS*, 2020.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.

[49] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-Deeplab: Stand-alone axial-attention for panoptic segmentation. In *ECCV*, 2020.

[50] Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning feature descriptors using camera pose supervision. In *ECCV*, 2020.

[51] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: Learned invariant feature transform. In *ECCV*, 2016.

[52] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *CVPR*, 2018.

[53] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning Two-View Correspondences and Geometry Using Order-Aware Network. *ICCV*, 2019.

[54] Zichao Zhang, Torsten Sattler, and Davide Scaramuzza. Reference Pose Generation for Long-term Visual Localization via Learned Features and View Synthesis. *IJCV*, 2020.