FeatureBooster: Boosting Feature Descriptors with a Lightweight Neural Network

Xinjiang Wang[1,2] Zeyu Liu[1,2] Yu Hu[1,2] Wei Xi[3] Wenxian Yu[1,2] Danping Zou[1,2]*
[1]Shanghai Key Laboratory of Navigation and Location Based Services, Shanghai Jiao Tong University
[2]SJTU SEIEE · G60 Yun Zhi AI Innovation and Application Research Center
[3]Intelligent Perception Institute, Midea Corporate Research Center
{wangxj83,ribosomal,henryhuyu,wxyu,dpzou}@sjtu.edu.cn xiwei1@midea.com

Abstract

We introduce a lightweight network to improve descriptors of keypoints within the same image. The network takes the original descriptors and the geometric properties of keypoints as the input, and uses an MLP-based self-boosting stage and a Transformer-based cross-boosting stage to enhance the descriptors. The boosted descriptors can be either real-valued or binary ones. We use the proposed network to boost both hand-crafted (ORB [34], SIFT [24]) and the state-of-the-art learning-based descriptors (SuperPoint [10], ALIKE [53]) and evaluate them on image matching, visual localization, and structure-from-motion tasks. The results show that our method significantly improves the performance of each task, particularly in challenging cases such as large illumination changes or repetitive patterns. Our method requires only 3.2ms on desktop GPU and 27ms on embedded GPU to process 2000 features, which is fast enough to be applied to a practical system. The code and trained weights are publicly available at github.com/SJTU-ViSYS/FeatureBooster.

1. Introduction

Extracting sparse keypoints or local features from an image is a fundamental building block in various computer vision tasks, such as structure from motion (SfM), simultaneous localization and mapping (SLAM), and visual localization. The feature descriptor, represented by a real-valued or binary descriptor, plays a key role in matching those keypoints across different images.

The descriptors are commonly hand-crafted in the early days. Recently, learning-based descriptors [10, 53] have shown to be more powerful than hand-crafted ones, especially in challenging cases such as significant viewpoint and illumination changes. Both hand-crafted and learning-based descriptors have shown to work well in practice. Some of them have become default descriptors for some applications. For example, the simple binary descriptor ORB [34] is widely used for SLAM systems [20, 29]. SIFT [24] is typically used in structure-from-motion systems.

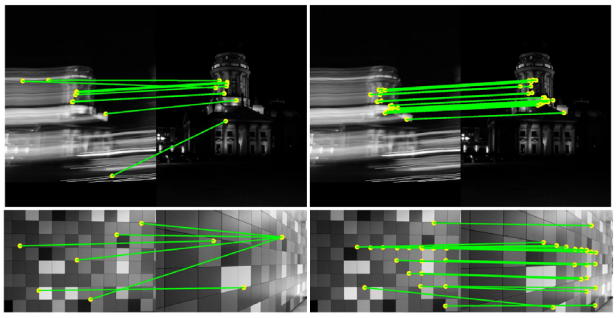


Figure 1. ORB descriptors perform remarkably better in challenging cases after being boosted by the proposed lightweight network. **Left column**: Matching results of using raw ORB descriptors. **Right column**: Results of using boosted ORB descriptors. Nearest neighbor search and RANSAC [14] were used for matching.

Considering that the descriptors have already been integrated into practical systems, replacing them with totally new ones can be problematic, as it may require more computing power that may not be supported by the existing hardware, or sometimes require extensive modifications to the software because of changed descriptor type (*e.g.* from binary to real).

In this work, we attempt to reuse existing descriptors and enhance their discrimination ability with as little computational overhead as possible. To this end, we propose a lightweight network to improve the original descriptors. The input of this network is the descriptors and the geomet-

*Corresponding Author: Danping Zou (dpzou@sjtu.edu.cn). This works was supported by National Key R&D Program (2022YFB3903802) and National of Science Foundation of China (62073214)

ric properties such as the 2D locations of all the keypoints within the entire image. Each descriptor is firstly processed by an MLP (Multi-layer perceptron) and summed with geometric properties encoded by another MLP. The new geometrically encoded descriptors are then aggregated by an efficient Transformer to produce powerful descriptors that are aware of the high-level visual context and spatial layout of those keypoints. The enhanced descriptors can be either real-valued or binary ones and matched by using Euclidean/Hamming distance respectively.

The core idea of our approach, motivated by recent work [25, 36, 41], is integrating the visual and geometric information of all the keypoints into individual descriptors by a Transformer. This can be better understood intuitively by considering when people are asked to find correspondences between images, they would check all the keypoints and the spatial layout of those keypoints in each image. With the help of the global receptive field in Transformer, the boosted descriptors contain global contextual information that makes them more robust and discriminative as shown in Fig. 1.

We apply our FeatureBooster to both hand-crafted descriptors (SIFT [24], ORB [34]) and the state-of-the-art learning-based descriptors (SuperPoint [10], ALIKE [53]). We evaluated the boosted descriptors on tasks including image matching, visual localization, and structure-from-motion. The results show that our method can significantly improve the performance of each task by using our boosted descriptors.

Because FeatureBooster does not need to process the image and adopts a lightweight Transformer, it is highly efficient. It takes only 3.2ms on NVIDIA RTX 3090 and 27ms on NVIDIA Jetson Xavier NX (for embedded devices) to boost 2000 features, which makes our method applicable to practical systems.

2. Related work

Feature descriptors: For a long time, the descriptors are commonly hand-crafted. SIFT [24] and ORB [34] are the most well-known hand-crafted descriptors, which are still widely used in many 3D computer vision tasks for their good performance and high efficiency. Hand-crafted descriptors are usually extracted from a local patch. It hence limits their representation capability on higher levels. With the development of deep learning and the emergence of patch dataset with annotation [7], learning-based descriptors have been widely studied. Most learning-based descriptors from patches adopt the network architecture introduced in L2-Net [44] and are trained with different loss functions, *e.g*. triplet loss [28, 43, 45], N-Pair loss [44] and list-wise ranking loss [17]. Learning-based dense descriptors [10, 12, 16, 30, 33, 50] can leverage information beyond local patches in that they are typically extracted from the entire image using convolutional neural networks, thus exhibiting superior performances on large viewpoint and illumination changes. Though a lot of descriptors have been invented, how to boost existing descriptors has received little attention, particularly through a learning-based approach.

Improve existing feature descriptors: It has been found that projecting existing descriptors into another space by a non-linear transformation leads to better matching results [32]. RootSIFT [2] shows that simply taking the square root of each element of the normalized SIFT descriptors can improve the matching results. Apart from improving the discrimination, some works also seek to compress the descriptors by reducing the descriptor's dimension, such as PCA-SIFT [19] and LDAHash [40]. A recent work [11] trained a network to map different types of descriptors into a common space such that different types of descriptors can be matched. Our work shares the core idea with this line of research but aims to enhance the discrimination ability to exist descriptors using a lightweight neural network.

Feature matching: Once feature descriptors are acquired, the correspondences between images are usually found by nearest neighbor (NN) search. The incorrect matches can be filtered by adopting some tricks (*e.g*. mutual check, Lowe's ratio test [24], and RANSAC [14]). However, NN search ignores the spatial and visual relationship between features and usually produces noisy matching results. To address this problem, SuperGlue [36] trained an attentional graph neural network by correlating two sets of local features from different images to predict the correspondences. Our approach is largely inspired by SuperGlue, but does not attempt to improve the matching process. It instead enhances the feature descriptors from a single image, such that a simple NN search can be used to produce competitive results. Therefore our approach can be seamlessly integrated into many existing pipelines such as a BoW(bag-of-word) [15] implementation.

Feature context: The distribution of feature locations and descriptors within an entire image forms a global context that can be helpful for feature matching as demonstrated in SuperGlue [36]. In this paper, we aim to integrate the global context information into original descriptors to boost their discrimination ability rather than learning to describe the image from scratch. The closest work to our approach is SConE [47] and ContextDesc [25]. SConE [47] develops a constellation embedding module to convert a set of adjacent features (including original descriptors and their spatial layout) into new descriptors. This module is designed for a particular type of descriptor (FREAK [1]). ContextDesc [25] uses two MLPs to encode the visual context and geometric context into global features to improve the local descriptors. It however requires to use of extra CNN to extract high-level features from the original image to construct the visual context.

By contrast, our method takes only the descriptors and geometric information (such as 2D locations) as the input and uses a lightweight Transformer to aggregate them to produce new descriptors. The new descriptors can be both binary or real-valued ones and can be seamlessly integrated into existing visual localization, SLAM, and structure-from-motion systems. No need to process the raw images makes our method very efficient and can run in real-time on embedded GPU devices.

3. Overview

We propose a lightweight network to boost the feature vectors (or descriptors) of a set of keypoints extracted from an image by some existing keypoint detectors as shown in Fig. 2. It takes only the feature descriptors as well as the geometric information such as feature position, orientation, and scale as the input, and outputs new descriptors that are much more powerful than the original ones. The new descriptors can be either real-valued or binary vectors which may be different from the original ones. Our feature booster does not need to process the image from which those keypoints are extracted, which makes our model lightweight and efficient, and can be more easily integrated into existing Structure-from-motion or SLAM systems. No need to access the original images also makes our approach possible to reuse 3D maps already built with certain types of features.

The proposed pipeline consists of two steps: **Self-boosting** and **Cross-boosting**. Self-boosting refers to using a lightweight MLP network to project the original feature vector into a new space. It also encodes geometric information such as 2D location, detection score, and orientation/scale to a high-dimensional vector to improve the descriptor. After that, cross-boosting explores the global context including the descriptors of other features and the spatial layout of all the features to further enhance the individual descriptors using a lightweight Transformer. The proposed network is trained end-to-end by using a loss function that consists of a ranking-based retrieval loss and an enhancement loss.

3.1. Self-boosting

For each keypoint i detected in the image, we can obtain its visual descriptor $\boldsymbol{d}_i$, a D dimensional real-valued or binary vector. The feature descriptors are then used to establish the correspondences between images by measuring their similarity. A powerful descriptor should be robust to the viewpoint and illumination changes to produce correct matching results. A lot of descriptors have been developed, including hand-crafted methods such as ORB [34], SURF [4], and SIFT [24], as well as more advanced learning-based methods such as SuperPoint [10]. However, there are still some problems with those descriptors.

For the hand-crafted ones, the first problem is that the similarity metric in the descriptor space is not optimal for feature matching. This has been noticed in [2], where a Hellinger distance is used to measure the SIFT's similarity instead using a Euclidean distance, which leads to a better matching performance. It can be seen from [32], changing the similarity metric is equivalent to projecting the original descriptors into another space. This motivates us to use an MLP (Multi-layer perceptron) to map the original descriptor into a new one.

MLP is a universal function approximator as shown by Cybenko's theorem [9]. Hence we can use an MLP to approximate the project function which we refer to as $\mathbf{MLP}_{desc}$. The transformed descriptor $\boldsymbol{d}_i^{tr}$ for keypoint i is the non-linear projection of the extracted descriptor $\boldsymbol{d}_i$:

$$\boldsymbol{d}_i^{tr} \leftarrow \mathbf{MLP}_{desc}(\boldsymbol{d}_i) \tag{1}$$

Given that the network's training phase is guided by a loss function with Euclidean or Hamming distance constraints, this MLP-based model enables the transformed descriptors to be well fit for measuring similarity in Euclidean or Hamming space respectively, especially for the hand-crafted descriptors. However, this projection hasn't exploited the geometric information of the key point which is valuable for matching [36]. Therefore, we also embed the geometric information into a high dimensional vector using another MLP ($\mathbf{MLP}_{geo}$) to further improve the descriptor. We encode not only the 2D location of keypoints (x_i, y_i), but also other information such as the scale s_i, orientation θ_i, and detection score c_i when they are available. The high-dimensional embedded geometric information is added to the transformed descriptor:

$$\boldsymbol{d}_i^{tr} \leftarrow \boldsymbol{d}_i^{tr} + \mathbf{MLP}_{geo}(\boldsymbol{p}_i). \tag{2}$$

Here, $\boldsymbol{p}_i = (x_i, y_i, c_i, \theta_i, s_i)$ represents all available geometric information as aforementioned.

3.2. Cross-boosting

Self-boosting enhances the descriptor of each keypoint independently without considering the possible correlation between different keypoints. For example, it does not exploit the spatial relationships between those keypoints, while the spatial contextual cues could greatly enhance the matching capability as demonstrated in [36]. Therefore, the boosted descriptors from the self-boosting stage are limited to the local context and still perform poorly under some challenging environments (*e.g.* repetitive patterns or weakly textured scenes). To address this issue, we further process those descriptors by a cross-boosting stage.

Motivated by SuperGlue [36], we use a Transformer to capture spatial contextual cues of the sparse local features extracted from the same image. We denote the Transformer

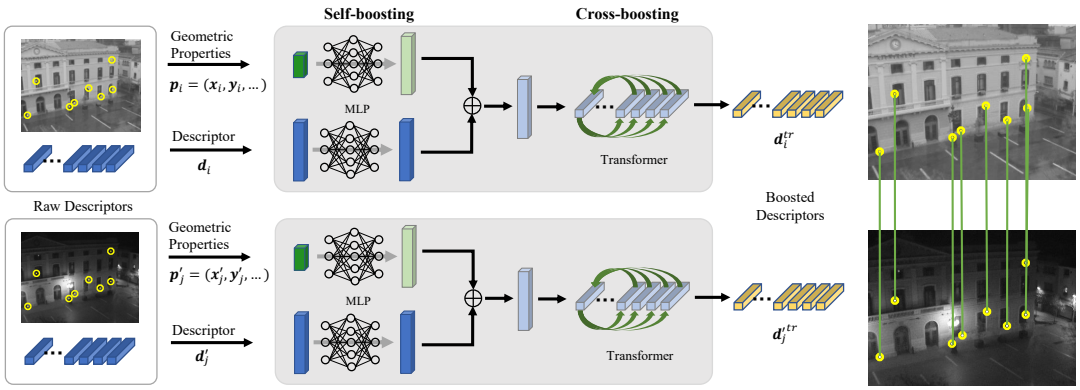


Figure 2. The proposed *FeatureBooster* pipeline consists of self-boosting and cross-boosting stages. Self-boosting applies an MLP to encode the geometric properties of a keypoint and combines it with a new descriptor projected by another MLP. In the cross-boosting stage, the geometrically encoded descriptors of all the keypoints within the entire image are then sent to a lightweight Transformer to generate boosted descriptors. Finally, the boosted descriptors are used for feature matching.

by **Trans** and the projection is described as:

$$(\boldsymbol{d}_1^{tr}, \boldsymbol{d}_2^{tr}, \dots \boldsymbol{d}_N^{tr}) \leftarrow \mathbf{Trans}(\boldsymbol{d}_1^{tr}, \boldsymbol{d}_2^{tr}, \dots \boldsymbol{d}_N^{tr}), \quad (3)$$

where the input of the Transformer is N local features within the same image, and the output is the enhanced feature descriptors. Compared with the MLP-based projection (see Eq. (1)), Transformer-based projection processes all the local features within the same image simultaneously. With the help of the attention mechanism in Transformer, all local features' information can be aggregated to form a global context. By integrating this global contextual information, the local feature descriptors may have larger receptive fields and adjust themselves according to their neighbors (or competitors in the case of feature matching). Therefore their distinguishability can be improved, especially for local features extracted from repetitive patterns as shown in Fig. 1.

The biggest issue of using a Transformer is that its attention mechanism requires high memory and computation costs. The transformer encoder layer consists of two sub-layers: an attention layer and a position-wise fully connected feed-forward network. The vanilla Transformer [49] uses a Multi-Head Attention (MHA) layer. Given an input $\mathbf{X} \in \mathbb{R}^{N \times D}$, where the i-th row is the D dimensional feature vector of keypoint i, the h-th head attention of $\mathbf{X}$ is defined as:

$$\begin{aligned} \mathbf{f}_h(\mathbf{X}) &= softmax(\frac{\mathbf{Q}_h \mathbf{K}_h^\top}{D_k})\mathbf{V}_h, \\ s.t.\ \mathbf{Q}_h &= \mathbf{X}\mathbf{W}_h^Q, \mathbf{K}_h = \mathbf{X}\mathbf{W}_h^K, \mathbf{V}_h = \mathbf{X}\mathbf{W}_h^V \end{aligned} \quad (4)$$

where $\mathbf{W}_h^Q \in \mathbb{R}^{D \times D_k}, \mathbf{W}_h^K \in \mathbb{R}^{D \times D_k}, \mathbf{W}_h^V \in \mathbb{R}^{D \times D_v}$ are the linear projections of for head h. Fig. 3(a) illustrates the computation graph of dot-product attention. The output of Multi-Head Attention is the concatenation of all the attention heads' outputs along the channel dimension.

MHA uses the attention matrix to enable the global interaction between query and value. The computation of the attention matrix relies on the matrix dot product between query and key, which results in a time and space complexity quadratic with the context size ($O(N^2D)$). It is easy to see that the complexity introduced by MHA makes Vanilla Transformer difficult to scale to inputs with a large context size (N). In our case, the context size (N) is the number of local features within an image. Unfortunately, it is very common that thousands of local features have been extracted within one image.

Attention-Free Transformer: To address the scalability problem in our case, we propose to use an efficient Attention-Free Transformer (specifically AFT-Simple) [51] to replace the MHA operation in a Vanilla Transformer. Unlike MHA or recent linearized attention [18], Attention-Free Transformer (AFT) does not use or approximate the dot product attention. Specifically, AFT rearranges the computation order of Q, K, and V, just like linear attention, but multiplies K and V element-wise instead of using matrix multiplication. The Attention-Free Transformer for keypoint i can be formulated as:

$$\begin{aligned} \mathbf{f}_i(\mathbf{X}) &= \sigma(\mathbf{Q}_i) \odot \frac{\sum_{j=1}^N \exp(\mathbf{K}_j) \odot \mathbf{V}_j}{\sum_{j=1}^N \exp(\mathbf{K}_j)} \\ &= \sigma(\mathbf{Q}_i) \odot \sum_{j=1}^N (softmax(\mathbf{K}) \odot \mathbf{V})_j \end{aligned} \quad (5)$$

where $\sigma(\cdot)$ is a Sigmoid function; $\mathbf{Q}_i$ represents i-th row of $\mathbf{Q}$; $\mathbf{K}_j, \mathbf{V}_j$ represent the j-th rows of $\mathbf{K}, \mathbf{V}$. AFT-simple performs a revised version of the MHA operation where the number of attention heads is equal to the model's feature dimension D and the similarity used in MHA is replaced by a kernel function $sim(\mathbf{Q}, \mathbf{K}) = \sigma(\mathbf{Q}) \cdot softmax(\mathbf{K})$. In

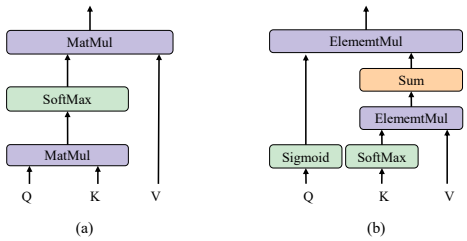


Figure 3. Different architectures of the attention layer. (a) Attention layer in a vanilla Transformer. (b) Attention-Free Transformer (AFT-simple), where only element-wise multiplication is required.

this way, attention can be computed by element-wise multiplication instead of matrix multiplication, which results in a time and space complexity that is linear with context and feature size ($O(ND)$). Fig. 3(b) illustrates the computation graph of AFT-Simple.

3.3. Loss Functions

As in previous work [17, 33], we treat the descriptor matching problem as nearest neighbor retrieval and use the Average Precision (AP) to train the descriptors. Considering transformed local feature descriptors $\mathbf{d^{tr}} = (\boldsymbol{d}_1^{tr}, \ldots, \boldsymbol{d}_N^{tr})$, we want to maximize the AP [6] for all descriptors and our goal for training is to minimize the following cost function:

$$\mathcal{L}_{AP} = 1 - \frac{1}{N}(\sum_{i}^{N} AP(\boldsymbol{d}_i^{tr})) \tag{6}$$

To ensure that the original descriptors will be boosted, we propose to use another loss to force the performance of transformed descriptors to be better than the original ones:

$$\mathcal{L}_{BOOST} = \frac{1}{N}\sum_{i}^{N} max(0, \frac{AP(\boldsymbol{d}_i)}{AP(\boldsymbol{d}_i^{tr})} - 1) \tag{7}$$

The final loss is the sum of the above two losses:

$$\mathcal{L} = \mathcal{L}_{AP} + \lambda\mathcal{L}_{BOOST} \tag{8}$$

where λ is a weight to regulate the second term. We use a differentiable approach (FastAP [8]) to compute the Average Precision (AP) for each descriptor.

Given a transformed descriptor $\boldsymbol{d}_i^{tr} \in \mathbb{R}^{1\times D}$ in the first image and the set of descriptors $\mathbf{d'}^{tr} \in \mathbb{R}^{N\times D}$ in the second image. FastAP can be computed by using the ground truth labels about matched pairs $\mathbf{M} = \{M^+, M^-\}$ and pairwise distance vector $Z \in \mathbb{R}^N$ with value domain Ω. By using distance quantization, Ω can be quantized as a finite set with b elements $\Omega = \{z_1, z_2, \ldots, z_b\}$, then the precision and recall can be reformulated as functions of the distance z:

$$\mathbf{Prec}(z) = P(M^+|Z < z) \tag{9}$$

$$\mathbf{Rec}(z) = P(Z < z|M^+) \tag{10}$$

where $P(M^+|Z < z)$ represents the prior distribution for positive matches M^+ conditioned on $Z < z$ and $P(Z < z|M^+)$ is the cumulative distribution function (CDF) for Z. Finally, the AP can be approximated by the area of precision-recall curve $\mathbf{PR}_z(\boldsymbol{d}_i^{tr}) = \{(\mathbf{Prec}(z), \mathbf{Rec}(z)), z \in \Omega\}$, which can be denoted as:

$$\mathbf{FastAP} = \int_{z\in\Omega} \mathbf{Prec}(z)d\mathbf{Rec}(z) \tag{11}$$

More details about FastAP are described in [8]. The ground truth labels about matches $\mathbf{M}$ can be acquired using the ground truth poses and depth maps. Note that the way to calculate distance vector Z is different for real-valued and binary descriptors.

3.4. Different types of descriptors

We are able to train our model to boost the descriptors into both binary and real-valued forms by using different ways to compute the distance vector Z.

Real-Valued Descriptors: We apply L_2 normalization to the output vector of the last layer of FeatureBooster, and the pairwise distance vector Z can be calculated as:

$$Z = 2 - 2\boldsymbol{d}_i^{tr}(\mathbf{d'}^{tr})^\top \tag{12}$$

In this case, the bound range of Z is $[0, 4]$ and we quantize the Ω as a finite set with 10 elements.

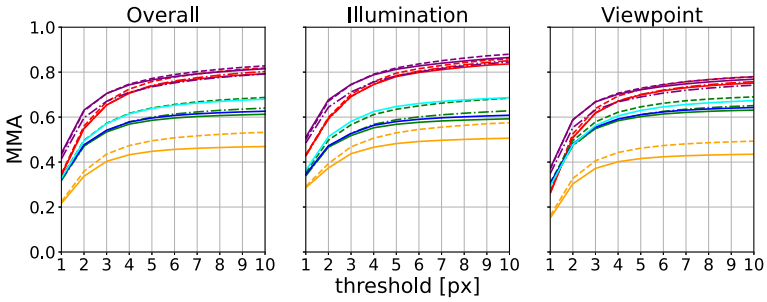
Binary Descriptors: We first use $tanh$ to threshold the output vector of the last layer of FeatureBooster to $[-1, 1]$. The output vector is then binarized to $\{-1, 1\}$. However, there is no real gradient defined for binarization. Our solution is to copy gradients from binarized vector to unbinarized vector following the straight-through estimator [5]. Finally, the pairwise distance vector Z can be obtained as:

$$Z = \frac{1}{2}(D - \boldsymbol{d}_i^{tr}(\mathbf{d'}^{tr})^\top) \tag{13}$$

For the Hamming distance, the values of Z are the integer in $\{0, 1, \ldots, D\}$, and AP can be computed in a closed form by setting $b = D$ in FastAP. However, we use $b = 10$ to get a larger margin between matching descriptors and non-matching descriptors as the discussion in [8].

4. Implementation details

In this section, we provide some implementation details for training FeatureBooster. FeatureBooster is plug-and-play and can be combined with any feature extraction process. In this paper, we trained FeatureBoosters for ORB [34], SIFT [24], SuperPoint [10], and ALIKE [53] respectively. We use ORB-SLAM2's [29] extractor for ORB extraction and COLMAP's [37, 39] extractor for SIFT extraction. For SuperPoint [10], we use its open-source repository



Method	Features	Matches	MMA @3 / @5
ORB [34]	2956	997	0.403 / 0.448
ORB+Boost-B (ours)	2956	1107	**0.436 / 0.495**
SIFT [24]	1675	772	0.534 / 0.586
SOSNet [45]	1675	797	0.571 / 0.638
RootSIFT [2]	1675	799	0.542 / 0.596
SIFT+Boost-F (ours)	1675	853	**0.573 / 0.640**
SIFT+Boost-B (ours)	1675	860	0.539 / 0.600
SuperPoint [10]	1562	884	0.654 / 0.738
SuperPoint+Boost-F (ours)	1562	920	**0.669 / 0.758**
SuperPoint+Boost-B (ours)	1562	911	0.654 / 0.741
ALIKE [53]	2578	1229	0.705 / 0.766
ALIKE+Boost-F (ours)	2578	1325	**0.705 / 0.772**
ALIKE+Boost-B (ours)	2578	1271	0.670 / 0.735

Figure 4. MMA curves in HPatches (the higher the better) and the number of matched points on average (the larger the better). The results show that our feature booster can improve the performance for all the features. Boost-F and Boost-B indicate real-valued boosted and binary boosted descriptors, respectively.

and the Non-Maximum Suppression (NMS) radius is 4 pixels. For ALIKE [53], we use its default open-source model.

Architecture details: All the models were implemented in PyTorch [31]. The Transformer in FeatureBooster uses $L = 9$ encoder layers for ALIKE and SuperPoint, and $L = 4$ for ORB and SIFT. The query, key, and value in the Transformer encoder have the same dimension D as that of the input descriptor. The feed-forward network in Transformer is an MLP with 2 layers where the output dimensions are $(2D, D)$. The geometric encoder is an MLP with five layers where the output dimensions are $(32, 64, 128, D, D)$ respectively. Note the 2D locations of keypoints are normalized by the largest image dimension and the feature orientation is represented in radians. For ORB (or binary) descriptors, we first convert them to a float vector and normalized them from $[0, 1]$ to $[-1, 1]$ and then send them to the 2-layer MLP with shortcut connection where the output dimensions are $(2D, D)$ like all other descriptors.

Training data: We trained all the FeatureBoosters on MegaDepth [21] and adopt the training scenes used in DISK [48]. We computed the overlap score between two images following D2-Net [12] and sampled 300 training pairs with an overlap score in $[0.1, 1]$ for each scene at every epoch. A random 512×512 patch centered around one correspondence is selected for each pair. During the training, all the local features were extracted on-the-fly, yielding up to 2048 local features from a single image. The labels for matched descriptors and unmatched descriptors were generated by checking the distance between the re-projected points and the keypoints. For matched descriptors, the distance is below 3 pixels. For unmatched descriptors, the distance is greater than 15 pixels, considering the possible annotation errors.

Training details: We set $\lambda = 10$ in the training loss and trained our FeatureBoosters using AdamW [23] optimizer. We increased the learning rate to 1×10^{-3} linearly in the first 500 steps and then decreased the learning rate in the form of cosine at each epoch in the following steps. The batch size is 16 during the training.

5. Experiments

After training our model on MegaDepth [21], we evaluate the trained model on image matching, visual localization, and structure-from-motion tasks using the public benchmark datasets. Note we do not fine-tune the model using the images from those datasets. We also show some matching results for real-world images from the Internet in Fig. 5. Finally, we also conduct an ablation study about the key components of our method.

5.1. Image Matching

We first evaluate our method on the image matching task using the HPatches [3] test sequences. HPatches dataset contains 116 different sequences of which 58 sequences have illumination changes and 58 sequences have viewpoint changes. Following D2Net [12], we excluded eight sequences for this experiment.

Experiment setup: We follow the evaluation protocol in D2Net [12] and record the mean matching accuracy (MMA) [27] under thresholds varying from 1 to 10 pixels, together with the numbers of features and matches. The MMA is defined as the average percentage of correct matches under different reprojection error thresholds. Like D2-Net, we use mutual nearest neighbor search as the matching method. For comparison, we report the results of raw descriptors, boosted descriptors by our approach, a variant for SIFT (RootSIFT [2]), and a learning-based patch descriptor (SOSNet [45]). All the DoG-based descriptors were computed from the same DoG keypoints for a fair comparison.

Result: Fig. 4 shows MMA results on HPatches under illumination and viewpoint change. Our method can enhance the performance of all descriptors for either the transformed real-valued descriptors or the binary ones. For SIFT, the transformed real-valued descriptors by our method outperforms SOSNet, while can find more correct matches as shown in the Table as shown in Fig. 4. In addition, we

Method	Aachen Day-Night V1.1 [52] (0.25m,2°) / (0.50m,5°) / (5.0m,10°) ↑		InLoc [42] (0.25m,10°) / (0.50m,10°) / (5.0m,10°) ↑	
	Day	Night	DUC1	DUC2
ORB [34]	80.6 / 87.9 / 93.6	31.9 / 37.2 / 49.2	24.7 / 33.3 / 42.4	26.7 / 37.4 / 44.3
ORB-Boost-B (Ours)	**83.1 / 89.8 / 94.7**	**49.2 / 61.8 / 73.3**	**35.4 / 50.5 / 59.1**	**38.9 / 51.9 / 61.8**
SIFT [24]	87.1 / 93.8 / 98.1	50.8 / 70.2 / 81.2	29.3 / 43.4 / 51.5	19.1 / 33.6 / 40.5
SOSNet [45]	**88.7 / 94.7 / 98.7**	58.1 / **78.5 / 92.7**	**35.9 / 50.0 / 64.6**	26.7 / **43.5 / 56.5**
RootSIFT [2]	86.8 / 94.1 / 98.4	57.1 / 76.4 / 88.5	30.3 / 46.5 / 57.1	22.1 / 42.7 / 50.4
SIFT+Boost-F (Ours)	87.1 / 94.5 / 98.1	62.3 / 78.0 / 92.1	31.8 / 43.9 / 57.1	24.4 / 36.6 / 49.6
SIFT+Boost-B (Ours)	87.5 / 94.5 / 98.1	**63.9** / 77.5 / 91.1	32.8 / 47.5 / 57.6	**30.5 / 43.5** / 51.1
SuperPoint [10]	87.9 / 94.3 / 98.2	67.0 / 84.8 / 95.8	36.9 / 57.6 / 64.6	38.2 / 55.0 / 65.6
SuperPoint+Boost-F (Ours)	**88.3 / 94.4 / 98.7**	**70.2 / 85.9 / 97.9**	**41.4 / 58.6 / 69.2**	**40.5 / 58.0 / 67.9**
SuperPoint+Boost-B (Ours)	87.4 / 94.1 / 97.9	68.6 / 84.8 / 96.3	36.9 / 54.5 / 65.7	35.9 / **58.0 / 67.9**
ALIKE [53]	**87.3** / 93.2 / 98.7	67.5 / 85.3 / 97.9	29.3 / 46.5 / 59.6	25.2 / 38.9 / 47.3
ALIKE+Boost-F (Ours)	86.7 / **94.2 / 99.0**	**72.8 / 86.9 / 98.4**	35.4 / 51.0 / 65.7	29.8 / 44.3 / 55.7
ALIKE+Boost-B (Ours)	86.9 / 93.8 / 98.3	71.7 / 86.4 / 96.9	**35.9 / 54.0 / 66.2**	**30.5 / 49.6 / 63.4**
SuperPoint+SuperGlue [10, 36]	89.6 / 96.4 / 99.3	73.3 / 90.6 / 100.0	44.9 / 64.6 / 78.3	49.6 / 73.3 / 77.1

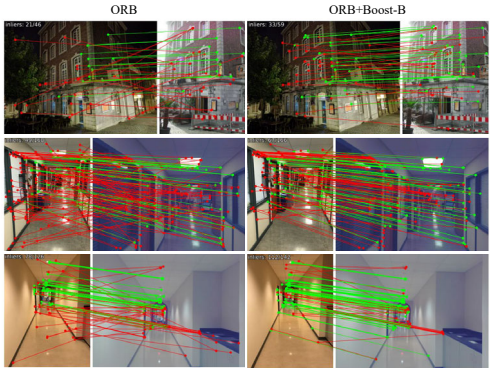


Table 1. Visual localization results in both outdoor (Aachen Day-Neight [52]) and indoor scenes (InLoc [42]). The positional and angular performances are present (the larger the better). Note that the boosted ORB (ORB-Boost-B) even outperforms ALIKE [53] and can compete with SuperPoint [10] in indoor scenes. Images on the right show some matching results before and after boosting using ORB [34] descriptors (red lines indicate wrong correspondences).

can see the potential of FeatureBooster for descriptor compression (real-valued descriptor to binary descriptor). The transformed binary descriptor from SuperPoint has a similar performance to the original SuperPoint under both illumination and viewpoint change while producing more correct matches. It is also interesting to see that the binary descriptor boosted from SIFT performs better than both SIFT and RootSIFT.

5.2. Visual Localization

In the second experiment, we evaluate our method in visual localization, a more complete pipeline in computer vision. Two challenging scenarios are selected for evaluation: an outdoor dataset with severe illumination changes and a large-scale indoor dataset with plenty of texture-less areas and repetitive patterns.

Experiment setup: For the outdoor scenes, we use the Aachen Day-Night dataset v1.1 [52], which contains 6697 day-time database images and 1015 query images (824 for the day and 191 for the night). For the indoor scenes, we use the InLoc dataset [42], which contains about 10k database images collected in two buildings. We use the hierarchical localization toolbox (HLoc) [35] for visual localization on Aachen Day-Night and InLoc dataset by replacing the feature extraction module with different feature detectors and descriptors. We use the evaluation protocol on the Long-Term Visual Localization Benchmark [46] and report the percentage of correct localized query images under given error thresholds. For comparison, we also report the result of the learning-based matching method (SuperPoint+SuperGlue). Not that all other methods use mutual nearest neighbor search for matching. We adopt ratio test or distance test for mutual nearest neighbor matching. For a fair comparison, the ratio or distance thresholds of all the transformed descriptors are selected according to the threshold criteria of their corresponding baselines[1].

Result: The results are shown in Tab. 1. Our method significantly improves the performance for all the features in both outdoor and indoor environments, especially for SIFT. After boosting, even the binary ORB descriptors can compete with the SuperPoint and outperform ALIKE in indoor environments (InLoc). We can see that the real-valued and binary boosted SIFT both show considerable competitiveness compared to SOSNet on the Day-Night outdoor dataset. The result also can show that SuperGlue still has the best performance in this experiment. However, our method boosts descriptors before the matching stage, making it more versatile and easy to insert into existing systems.

5.3. Structure-from-motion

Experiment setup: We use three medium-scale datasets in the ETH SfM benchmark [38] following D2-Net [12] for evaluation. We use exhaustive image matching for all these datasets and adopt ratio test or distance test for mutual nearest neighbor matching. Then, we run the SfM using COLMAP [37, 39]. Following the evaluation protocol defined by [38], we report the number of registered images, sparse points, total observations in image, mean feature track length, and mean re-projection error.

Result: Tab. 2 shows the results. Our approach again enhances the performance of all the features on the task of structure-from-motion. Our method can help the original features to produce a more complete reconstruction, as our approach can register more images and reconstruct more 3D points as shown in Tab. 2. Besides, our FeatureBooster can achieve higher feature track length, which means that we can find more correspondences between images to reconstruct 3D points while tracking the same features across more images. We also observe the situation that has been

[1] Please see the supplementary material for additional details.

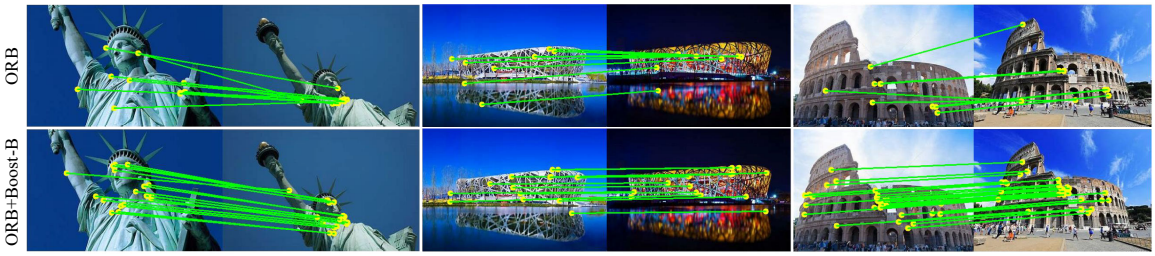


Figure 5. Matching results of using the original ORB [34] descriptors (**Top row**) and the boosted ORB descriptors (**Bottom row**) for Internet images. Nearest neighbor search and RANSAC [14] were applied for matching.

Dataset	Descriptor	#Reg. Images ↑	#Sparse. Point ↑	#Obs. ↑	#Track Length ↑	#Reproj. Error ↓
Madrid Metropolis 1344 images	SIFT [24]	417	29653	210460	7.10	**0.78px**
	SOSNet [45]	**464**	**35288**	**260737**	**7.39**	0.87px
	RootSIFT [2]	443	32613	230487	7.07	0.79px
	SIFT+Boost-B (ours)	415	34497	242053	7.02	0.86px
	SIFT+Boost-F (ours)	409	30020	221320	7.37	0.88px
	SuperPoint [10]	512	29131	230966	7.93	**1.14px**
	SuperPoint+Boost-B (ours)	433	25872	218370	**8.44**	1.18px
	SuperPoint+Boost-F (ours)	**534**	**34033**	**276204**	8.12	1.19px
Gendarmenmarkt 1463 images	SIFT [24]	944	75369	476495	6.32	**0.91px**
	SOSNet [45]	**972**	85507	**591623**	**6.92**	1.00px
	RootSIFT [2]	955	77888	511209	6.56	0.93px
	SIFT+Boost-B (ours)	944	**95537**	581878	6.09	0.99px
	SIFT+Boost-F (ours)	937	84496	552081	6.53	1.01px
	SuperPoint [10]	997	70971	535761	7.55	**1.18px**
	SuperPoint+Boost-B (ours)	951	62426	513442	**8.22**	1.23px
	SuperPoint+Boost-F (ours)	**1044**	**84052**	**635591**	7.56	1.20px
Tower of London 1576 images	SIFT [24]	667	61906	457193	7.39	**0.78px**
	SOSNet [45]	**738**	71734	**558944**	**7.79**	0.84px
	RootSIFT [2]	674	62348	472817	7.58	0.79px
	SIFT+Boost-B (ours)	690	**73954**	515206	6.97	0.82px
	SIFT+Boost-F (ours)	681	66309	491273	7.41	0.83px
	SuperPoint [10]	712	38921	313825	8.06	**1.12px**
	SuperPoint+Boost-B (ours)	653	34641	290505	**8.39**	1.14px
	SuperPoint+Boost-F (ours)	**773**	**45687**	**360642**	7.89	1.14px

Table 2. Results on structure-from-motion. Our method improves the performance of the existing descriptors (SIFT [24], and SuperPoint [10]) in three datasets of ETH SfM benchmark [38].

Descriptor	Self Boosting		Cross Boosting	HPatches Matches	HPatches MMA @3 / @5
	MLP_{desc}	MLP_{geo}			
SuperPoint [10]				883	0.654 / 0.738
	✓			883	0.654 / 0.738
	✓	✓		884	0.655 / 0.739
	✓		✓	893	0.657 / 0.742
	✓	✓	✓	919	**0.669 / 0.758**

Table 3. Ablation study on SuperPoint [10] in HPatches [3] (the higher the better). The results show that cross-boosting can significantly improve the performance.

discussed in [26, 45] that more matches tend to lend higher re-projection error, and we think this issue can be addressed by recent work on keypoint position refinement [13, 22].

5.4. Ablation Study

Tab. 3 shows an ablation study of different components in our network. The study shows that geometric encoding is necessary for self-boosting, and the cross-boosting has a better performance for descriptor boosting. With the help of both modules, our transformed descriptors perform significantly better.

6. Discussion

Computational cost: Our network is lightweight and efficient. We measure the runtime of our method on both a desktop GPU and an embedded GPU. A forward pass with 2000 features in NVIDIA RTX 3090 takes on average 3.2/4.7ms for our 4/9 layers network, while in NVIDIA Jetson Xavier NX it needs 27/46ms.

Generalization: Though for each feature we need to train their corresponding FeatureBooster, experiments show that our approach works well for various classes of descriptors (hand-crafted or learned, binary or real-valued). Our models are trained with the MegaDepth [21] dataset and do not need to be fine-tuned for different tasks or datasets.

Limitations: The performance of the boosted descriptor is limited by the representation ability of the raw descriptor, though the performance gain tends to be larger for weaker descriptors like ORB. Our approach cannot be applied to enhance dense features because the computational cost grows with the number of feature points.

7. Conclusion

We introduce a descriptor enhancement stage into the traditional feature matching pipeline and propose a versatile and lightweight framework for descriptor enhancement called FeatureBooster. FeatureBooster jointly processes the geometric properties and visual descriptors of all the keypoints within a single image to extract the global contextual information. With the help of the global context, the transformed descriptors become powerful even though the original descriptor is very weak. Our experiments show that FeatureBooster can help various classes of descriptors (SIFT, ORB, SuperPoint, and ALIKE) to perform better under different vision tasks. Furthermore, our FeatureBooster demonstrates its potential for descriptor compression and can run in real time. We believe that our FeatureBooster can be useful for many practical applications.

References

[1] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. FREAK: Fast retina keypoint. In *CVPR*, pages 510–517, 2012. 2

[2] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, pages 2911–2918, 2012. 2, 3, 6, 7, 8

[3] 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, 8

[4] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *ECCV*, pages 404–417, 2006. 3

[5] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. 5

[6] Kendrick Boyd, Kevin H Eng, and C David Page. Area under the precision-recall curve: point estimates and confidence intervals. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 451–466. Springer, 2013. 5

[7] Matthew Brown, Gang Hua, and Simon Winder. Discriminative learning of local image descriptors. *IEEE TPAMI*, 33(1):43–57, 2010. 2

[8] Fatih Cakir, Kun He, Xide Xia, Brian Kulis, and Stan Sclaroff. Deep metric learning to rank. In *CVPR*, pages 1861–1870, 2019. 5

[9] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989. 3

[10] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, pages 224–236, 2018. 1, 2, 3, 5, 6, 7, 8

[11] Mihai Dusmanu, Ondrej Miksik, Johannes L Schönberger, and Marc Pollefeys. Cross-descriptor visual localization and mapping. In *ICCV*, pages 6058–6067, 2021. 2

[12] 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. 2, 6, 7

[13] Mihai Dusmanu, Johannes L Schönberger, and Marc Pollefeys. Multi-view optimization of local feature geometry. In *ECCV*, pages 670–686, 2020. 8

[14] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 1, 2, 8

[15] Dorian Gálvez-López and Juan D Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012. 2

[16] Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. S2Dnet: learning image features for accurate sparse-to-dense matching. In *ECCV*, pages 626–643, 2020. 2

[17] Kun He, Yan Lu, and Stan Sclaroff. Local descriptors optimized for average precision. In *CVPR*, pages 596–605, 2018. 2, 5

[18] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *ICML*, pages 5156–5165, 2020. 4

[19] Yan Ke and Rahul Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *CVPR*, pages II–II, 2004. 2

[20] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *Int. J. Robot. Res.*, 34(3):314–334, 2015. 1

[21] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, pages 2041–2050, 2018. 6, 8

[22] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-Perfect Structure-from-Motion with Featuremetric Refinement. In *ICCV*, pages 5987–5997, 2021. 8

[23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[24] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1, 2, 3, 5, 6, 7, 8

[25] 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. In *CVPR*, pages 2527–2536, 2019. 2

[26] Zixin Luo, Tianwei Shen, Lei Zhou, Siyu Zhu, Runze Zhang, Yao Yao, Tian Fang, and Long Quan. GeoDesc: Learning local descriptors by integrating geometry constraints. In *ECCV*, pages 168–183, 2018. 8

[27] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE TPAMI*, 27(10):1615–1630, 2005. 6

[28] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. *NeurIPS*, 30, 2017. 2

[29] Raul Mur-Artal and Juan D Tardós. ORB-SLAM2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017. 1, 5

[30] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. LF-Net: Learning local features from images. *NeurIPS*, 31, 2018. 2

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019. 6

[32] James Philbin, Michael Isard, Josef Sivic, and Andrew Zisserman. Descriptor learning for efficient retrieval. In *ECCV*, pages 677–691, 2010. 2, 3

[33] Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noe Pion, Gabriela Csurka, Yohann Cabon, and Martin Humen-

berger. R2D2: repeatable and reliable detector and descriptor. *arXiv preprint arXiv:1906.06195*, 2019. 2, 5

[34] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *ICCV*, pages 2564–2571, 2011. 1, 2, 3, 5, 6, 7, 8

[35] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In *CVPR*, 2019. 7

[36] 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, 3, 7

[37] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113, 2016. 5, 7

[38] Johannes L Schonberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *CVPR*, pages 1482–1491, 2017. 7, 8

[39] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, pages 501–518, 2016. 5, 7

[40] Christoph Strecha, Alex Bronstein, Michael Bronstein, and Pascal Fua. LDAHash: Improved matching with smaller descriptors. *IEEE TPAMI*, 34(1):66–78, 2011. 2

[41] 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

[42] 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*, pages 7199–7209, 2018. 7

[43] Yurun Tian, Axel Barroso Laguna, Tony Ng, Vassileios Balntas, and Krystian Mikolajczyk. HyNet: Learning local descriptor with hybrid similarity measure and triplet loss. *NeurIPS*, 33:7401–7412, 2020. 2

[44] Yurun Tian, Bin Fan, and Fuchao Wu. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *CVPR*, pages 661–669, 2017. 2

[45] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. SOSNet: Second order similarity regularization for local descriptor learning. In *CVPR*, pages 11016–11025, 2019. 2, 6, 7, 8

[46] 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. *IEEE TPAMI*, 2020. 7

[47] Tomasz Trzcinski, Jacek Komorowski, Lukasz Dabala, Konrad Czarnota, Grzegorz Kurzejamski, and Simon Lynen. SConE: Siamese constellation embedding descriptor for image matching. In *ECCVW*, pages 0–0, 2018. 2

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

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017. 4

[50] Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning feature descriptors using camera pose supervision. In *ECCV*, pages 757–774, 2020. 2

[51] Shuangfei Zhai, Walter Talbott, Nitish Srivastava, Chen Huang, Hanlin Goh, Ruixiang Zhang, and Josh Susskind. An attention free transformer. *arXiv preprint arXiv:2105.14103*, 2021. 4

[52] 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

[53] Xiaoming Zhao, Xingming Wu, Jinyu Miao, Weihai Chen, Peter CY Chen, and Zhengguo Li. ALIKE: Accurate and Lightweight Keypoint Detection and Descriptor Extraction. *IEEE TMM*, 2022. 1, 2, 5, 6, 7