Computational Visual Media
https://doi.org/10.1007/s41095-023-0363-3 Vol. 10, No. 5, October 2024, 873–884

Research Article

FilterGNN: Image feature matching with cascaded outlier filters and linear attention

Jun-Xiong Cai[1], **Tai-Jiang Mu**[1] (✉), **and Yu-Kun Lai**[2]

Abstract The cross-view matching of local image features is a fundamental task in visual localization and 3D reconstruction. This study proposes FilterGNN, a transformer-based graph neural network (GNN), aiming to improve the matching efficiency and accuracy of visual descriptors. Based on high matching sparseness and coarse-to-fine covisible area detection, FilterGNN utilizes cascaded optimal graph-matching filter modules to dynamically reject outlier matches. Moreover, we successfully adapted linear attention in FilterGNN with post-instance normalization support, which significantly reduces the complexity of complete graph learning from $O(N^2)$ to $O(N)$. Experiments show that FilterGNN requires only 6% of the time cost and 33.3% of the memory cost compared with SuperGlue under a large-scale input size and achieves a competitive performance in various tasks, such as pose estimation, visual localization, and sparse 3D reconstruction.

Keywords image matching; transformer; linear attention; visual localization; sparse reconstruction

1 Introduction

Finding pixel-wise correspondences in image pairs is an essential step in camera pose estimations and has been widely used for structure-from-motion (SfM) [1], simultaneous localization and mapping (SLAM) [2, 3], and visual localization [4] purposes. Most of the existing methods require two phases: local feature extraction and feature matching. Over the past decade, significant effort [5, 6] has been devoted to feature extraction using deep convolutional neural networks (DCNNs). Recently, some transformer-based [7] methods [8–13] have been proposed to significantly improve the matching ability compared with the traditional nearest neighbor (NN) searching strategy. However, some additional computational costs challenge their practical use in real-time applications.

Attention-based graph neural networks, such as SuperGlue [8], primarily benefit from the transformer's support for irregular data and the aggregated global context through a pairwise attention mechanism. In particular, (a) self-attention, which exhaustively calculates the correlation between any two keypoints extracted from the same image, is used to aggregate the inner-view global context. (b) Correspondingly, cross attention is applied to a complete bipartite graph comprising two keypoint sets grouped by source images to learn cross-view information. (c) Unlike bipartite graph matching, the local feature matching task involves numerous unmatchable keypoints. Therefore, a reasonable rejection mechanism is required to detect the optimal matching layer. In refining local descriptors using complete graph-based attention, SuperGlue achieves significant performance gains in many pose estimation [14, 15] and visual localization benchmarks [16–18]. However, fully connected attention mechanisms (as shown in Fig. 1(c)) result in a computational complexity of $O(N^2d)$[①], which is

1 Key Laboratory of Pervasive Computing, Ministry of Education, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China. E-mail: J.-X. Cai, caijunxiong000@163.com; T.-J. Mu, taijiang@tsinghua.edu.cn (✉).

2 School of Computer Science and Informatics, Cardiff University, Wales CF24 4AG, UK. E-mail: yukun.lai@cs.cardiff.ac.uk.

Manuscript received: 2023-02-26; accepted: 2023-06-23

① N denotes the input size; and d, the number of feature dimensions.

significantly higher than conducting an NN search.

Efforts to reduce the attention computation have been made. One method involves building a sparse graph from the inputs. Owing to the discrete and unordered nature of local image features, traditional methods such as spatial neighbor attention [19–21], which rely on ordered sliding windows, cannot be directly applied. More appropriate solutions focus on building subgraphs by sampling [9], projection [22, 23], or clustering [10, 24], thereby reducing the computational complexity to $O(kNd)$, where k is a small constant. Typically, these methods inevitably result in losing information in the size dimension, and the ratio of k to N should be selected carefully. A second method involves designing linear kernel functions to approximate fully connected attention [23, 25, 26]. However, previous work on ClusterGNN [10] reported the incompatibility of linear approximations with cross attention, resulting in a significant drop in the matching accuracy.

This study proposes FilterGNN, which effectively combines the two aforementioned methods for the comprehensive optimization of feature matching. Previous studies [8, 10] detected unmatchable points only in the final optimal matching layer. Conversely, FilterGNN exploits hierarchical outlier filters to dynamically reject invalid outliers interspersed between attentional aggregation blocks. As shown in Fig. 1, the low-level filter can quickly reject isolated keypoints outside the covisible area according to the basic descriptors, and the high-level filter accurately removes outlier matches according to the refined descriptors considering sparsity, visibility, and geometric distribution consistency.

To further accelerate the matching, we focused on approximating the fully connected attention with linear attention by considering the following aspects. (i) The residual attention block of SuperGlue [8] may lead to excessive variance amplification, which affects the convergence speed and performance. Therefore, a reasonable normalization layer is recommended, particularly for training from scratch. (ii) Considering the potentially large domain gaps between the input image pairs, such as orientation, scale, and lighting, we adopted instance normalization [27] instead of layer normalization [28] in the vanilla transformer [7]. (iii) The training gradient for linear attention is not as good as that for standard attention. Because the attention layer does not require additional learnable parameters, rather than training from scratch, we finetuned the linear attention using pretrained weights from standard attention. This preserves a high matching performance and helps the training converge faster.

The main contributions of this study are summarized as follows:

(1) We propose a cascaded optimal graph matching filter module that can dynamically reject unmatchable keypoints. It reduces the computational cost and provides a better feature

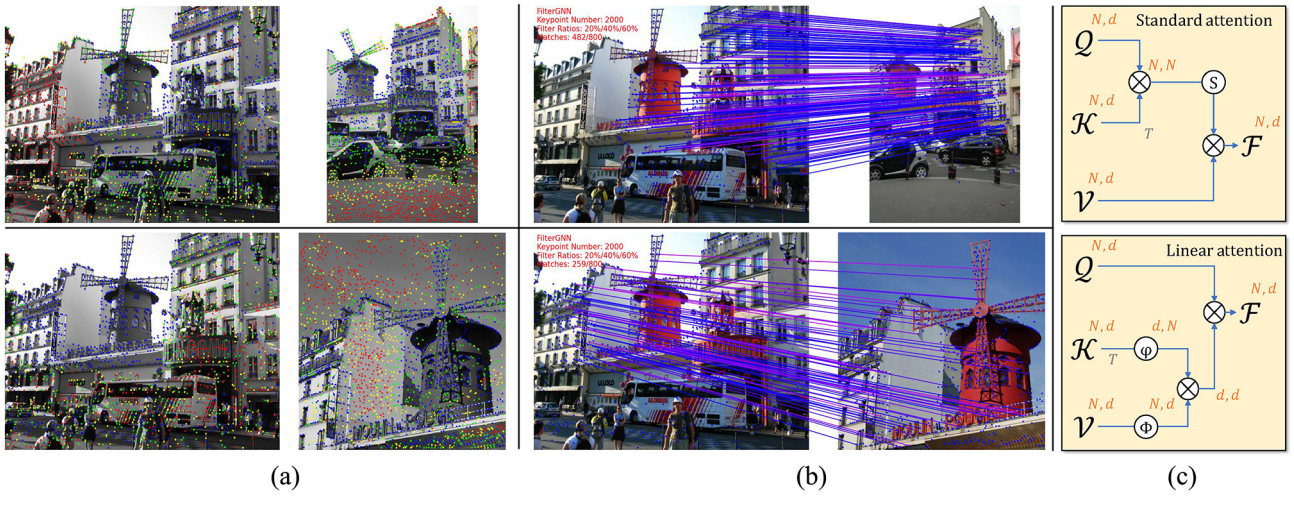


Fig. 1 (a) Visualization of the proposed hierarchical filtering. Keypoints in red, yellow, and green are discarded layer-by-layer. (b) The remaining keypoints (blue) are used for the final feature matching with our FilterGNN. (c) Standard and linear attention. $\mathcal{Q} \otimes \mathcal{K}^{\mathrm{T}}$ yields a computational complexity of $O(N^2 d)$, where $\otimes$ indicates matrix multiplication and (ϕ, φ) indicates kernel functions with linear complexity. The dimensions of the matrices are shown in orange.

distribution space for highly correlated keypoints.

(2) We propose an efficient and effective linear GNN architecture with post instance normalization, significantly reducing the computational complexity from $O(N^2d)$ to $O(Nd)$.

(3) Extensive experiments on various computer vision tasks demonstrate the applicability of our method, which achieves competitive results compared with state-of-the-art (SOTA) methods and demonstrates a significantly higher efficiency.

2 Related work

Image feature matching. Traditional pipelines primarily focus on robust interest-point detection and visual descriptor computations. SIFT [29] is a scale-invariant handcrafted feature method widely used in pose estimation tasks of SfM and multiview stereo (MVS). ORB [30] focuses on efficiency and is primarily applied to SLAM. Recently, deep convolutional neural networks (CNNs) have inspired many learning-based image feature extraction methods such as D2Net [5], SuperPoint [6], and ASLFeat [31], whose matching ability significantly surpasses that of handcrafted methods. They describe the content of local regions and often perform data augmentation on the scale, rotation, and imaging perturbations to achieve orientation invariance [32] or affine invariance [33].

Recently, Sarlin et al. [8] proposed SuperGlue, which exploits a transformer-based graph neural network to aggregate inner-global and cross-view information from keypoint sets of image pairs. As feature-matching networks aim to refine the descriptors, the source and target image features can be used as inputs directly. SuperGlue improves the transformer architecture of the encoder (stacked self-attention modules) and decoder (stacked cross-attention modules) with alternately stacked self- and cross-attention modules. However, SuperGlue suffers from a quadratic computational complexity of $O(N^2d)$, which is impractical for direct application in real-time systems.

Efficient transformer-based architecture. The transformer architecture has succeeded in both natural language processing [34] and computer vision tasks [35]. The inputs (text or images) for these tasks have regular structures, from which sparse graph patterns can be easily built. The sparse transformer [19, 36] performs attention computation only for the text subsequences within a shifted local block. Similarly, SwinTransformer [37] adopts a sliding window mechanism to compute the multi-level local attention efficiently using the high sparsity of the local window. Linformer [22] performs linear projections on the dimension of the input size, which requires ordering the input elements. These methods cannot be directly generalized to cross-image matching tasks because no reasonable manner of predefining the order and spatial adjacency of the sparse input keypoints exists.

SGMNet [9] simulates seed downsampling through attentional pooling with a computational complexity of $O(kNd)$, which is affected by the number of seeds k, particularly for large-scale inputs. Following SGMNet, Suwanwimolkul and Komorita [12] proposed neighborhood attention with an additional pairwise neighbor layer. ClusterGNN [10] and RoutingTransformer [38] divide the complete graph into multiple subgraphs in terms of semantic similarity by clustering and then perform only self-attention within each subgraph. The ideal time complexity of a GNN is $O(N^{1.5}d)$. These methods focus on building appropriate sparse graphs from a complete input graph to simplify attention computation.

Katharopoulos et al. [25] proposed a linear approximation of the attention layer using a kernel function. As shown in Fig. 1, the computational complexity is reduced by changing the order of the matrix multiplication. Several follow-up studies [23, 26, 39] have designed different types of kernel functions for different tasks. We drew inspiration from these kernel-function-approximation-based methods to make significant progress in image-matching tasks.

3 FilterGNN architecture

3.1 Overview

Given two keypoint sets $(\mathcal{X}_\mathrm{A}, \mathcal{X}_\mathrm{B})$ from a pair of images (A, B), feature matching is used to determine the correspondences $\mathcal{M} = \{(i, j)\}$ that make the 3D positions of the features $\mathcal{X}_\mathrm{A}^{(i)}$ and $\mathcal{X}_\mathrm{B}^{(j)}$ as close as possible. Input $\mathcal{X}$ comprises keypoint descriptors $\mathcal{D}$ and positions $\mathcal{P} = \{(u, v, c)\}$, where (u, v) are the image coordinates, and c denotes the corresponding keypoint detection score.

Typically, we can build a similarity score matrix directly using the cosine distances of the visual descriptors and generate correspondences through an NN search. Currently, most visual descriptors such as SIFT [29], SuperPoint [6], and D2net [5] encode only the local context. However, in long-term visual localization systems, many interference factors exist, such as lighting conditions, camera poses, and repetitive structures (buildings, checkerboards, etc.), that cannot be effectively handled using local descriptors.

Therefore, the proposed FilterGNN aims to refine local visual descriptors for feature matching. As shown in Fig. 2, FilterGNN applies optimal GMFs to the keypoint sets in a cascaded manner to filter out unmatchable keypoints step-by-step, in addition to refining the descriptors by aggregating the global context through an attention cascade, such as SuperGlue. Each GMF is dedicatedly designed with a novel linear attention GNN. Therefore, the final correspondences can be efficiently and accurately generated by performing a traditional NN search on the remaining distilled keypoint sets.

3.2 Graph matching filter (GMF)

Similar to the lightweight SuperGlue, our GMF comprises a shared positional encoder, a linear attention GNN, and a top-K filter. The positional encoder is used to extract the geometric content, and the linear attention GNN was designed to aggregate global information. After refining the input feature descriptors, we removed a fixed proportion of unmatchable points based on their cross-image-matching scores.

3.2.1 Positional encoder

The positional encoder module is defined as Eq. (1):

$$\widetilde{\mathcal{F}} = \sigma(\mathcal{D} \oplus \mathrm{mlp}_{\mathrm{pe}}(\mathcal{P})) \tag{1}$$

where σ indicates instance normalization [27], and $\oplus$ denotes matrix addition. $\mathcal{D}$ contains feature descriptors. $\mathrm{mlp}_{\mathrm{pe}}$ is the position encoding multi-layer perceptrons (MLPs) that perform the high-dimensional embedding of input image coordinates $\mathcal{P}$. Here, instance normalization was used to control the variance of the output features.

3.2.2 Linear attention GNN

A linear attention GNN comprises L alternating self- and cross-attention blocks. Figures 3(a) and 3(b) illustrate this. Unlike SuperGlue, we placed additional normalization layers after residual computing. Inspired by the SwinTransformerV2 [41], we demonstrate the variance change in Fig. 3.

The feedforward layer is simply defined as a residual MLP $\mathrm{mlp}_{\mathrm{ff}}$ as Eq. (2):

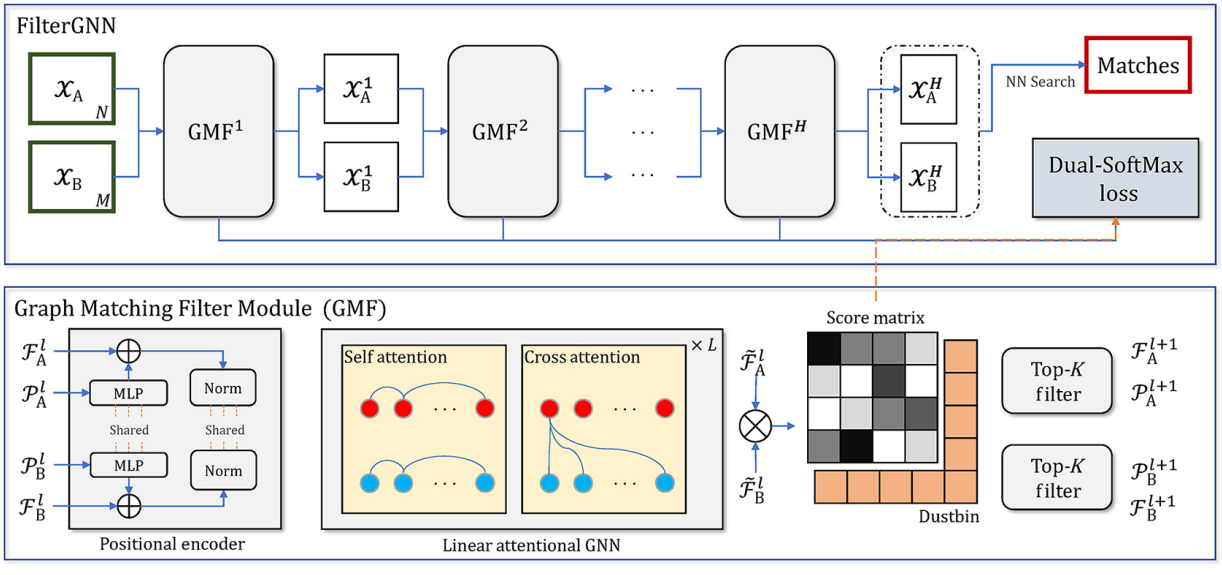


Fig. 2 FilterGNN architecture. FilterGNN uses cascaded optimal graph matching filter (GMF) modules (Section 3.2) to efficiently filter isolated keypoints for improved feature matching. Each GMF module, acting as a miniature SuperGlue [8] block, refines the descriptors $\mathcal{F}$ of the input keypoints set $\mathcal{X} = (\mathcal{F}, \mathcal{P})$ using a shared positional encoder (Section 3.2.1) and L stacked linear attention GNN layers (Section 3.2.2). A top-K filter is adopted to remove the k keypoints with the lowest matching probabilities, gradually reducing the size of the keypoint set. The score matrix of each GMF is then expanded with an additional dustbin dimension to detect outliers, and it generates the matching loss using the dual-softmax operator [40] for training. Final correspondences are generated by performing the traditional NN search on the remaining, distilled keypoint sets $(\mathcal{X}_{\mathrm{A}}^{H}, \mathcal{X}_{\mathrm{B}}^{H})$ after H GMFs.

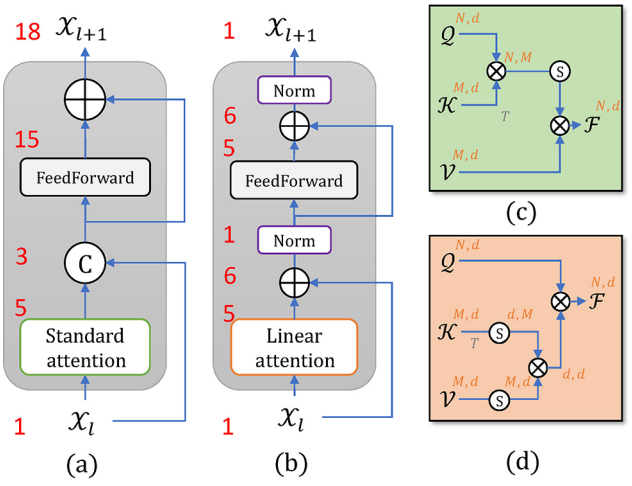


Fig. 3 (a) Original attention block in SuperGlue. (b) Our proposed linear attention block with post-norm. (c) Standard attention. (d) Linear attention. The red numbers indicate the variance during the operation. The variance multipliers of both the feedforward and attention layers are assumed to be five.

$$\mathrm{FF}(X) = \sigma(X \oplus \mathrm{mlp}_{\mathrm{ff}}(X)) \tag{2}$$

For self/cross-attention block $\widetilde{\mathrm{Att}}$, source features $\mathcal{F}_{\mathrm{src}} \in \mathbb{R}^{N,d}$ and target feature $\mathcal{F}_{\mathrm{tgt}} \in \mathbb{R}^{M,d}$ refine the aggregated features as inputs and outputs, respectively, by adding the attention output to $\mathcal{F}_{\mathrm{src}}$. Here, M and N denote the numbers of remaining keypoints in the two images not required to be equal. $\widetilde{\mathrm{Att}}$ are defined as Eq. (3):

$$\widetilde{\mathrm{Att}}(\mathcal{F}_{\mathrm{src}}, \mathcal{F}_{\mathrm{tgt}}) = \sigma(\mathcal{F}_{\mathrm{src}} \oplus \mathrm{Att}(\mathcal{Q}, \mathcal{K}, \mathcal{V}) \otimes W_{\mathrm{att}})$$
$$\begin{cases} \mathcal{Q} = \mathcal{F}_{\mathrm{src}} \otimes W_Q \\ \mathcal{K} = \mathcal{F}_{\mathrm{tgt}} \otimes W_K \\ \mathcal{V} = \mathcal{F}_{\mathrm{tgt}} \otimes W_V \end{cases} \tag{3}$$

where $W_{\mathrm{att},Q,K,V} \in \mathbb{R}^{d,d}$ denotes the linear projection weights, and $\otimes$ denotes matrix multiplication. $\mathcal{Q}$, $\mathcal{K}$, and $\mathcal{V}$ correspond to the queries, keys, and values in the transformer architecture, respectively. When the source and target are the same (e.g., from the same image), $\widetilde{\mathrm{Att}}$ performs self-attention; otherwise, it performs cross-attention.

The attention mechanism is at the core of the proposed GNN. As shown in Fig. 3(c), for the standard attention mechanism of SuperGlue [8] from the vanilla transformer [7], $\mathrm{Att}(\mathcal{Q}, \mathcal{K}, \mathcal{V})$ is defined as

$$\mathrm{Att}_{\mathrm{std}}(\mathcal{Q}, \mathcal{K}, \mathcal{V}) = \mathrm{Softmax}(\frac{\mathcal{Q} \otimes \mathcal{K}^{\mathrm{T}}}{\sqrt{d}}) \otimes \mathcal{V} \tag{4}$$

where the computational complexity of $\mathcal{Q} \otimes \mathcal{K}^{\mathrm{T}}$ is $O(N^2 d)$, which is the bottleneck of the entire method.

Linear attention [25] through the kernel function approximation is defined as Eq. (5):

$$\mathrm{Att}_{\mathrm{linear}}(\mathcal{Q}, \mathcal{K}, \mathcal{V}) = \phi(\mathcal{Q}) \otimes (\varphi(\mathcal{K}^{\mathrm{T}}) \otimes \mathcal{V})$$
$$\phi(x) = \varphi(x) = \mathrm{Softmax}(x) \tag{5}$$

In addition, (ϕ, φ) has the following options. The definition in the equation above was borrowed from that of efficient attention [39]. The computational complexity of $\mathrm{Att}_{\mathrm{linear}}$ is $O(Nd^2)$. Typically, N is significantly greater than d. Therefore, $O(N^2 d)$ and $O(Nd^2)$ can be represented as $O(N^2)$ and $O(N)$, respectively.

As reported by Shi et al. [10], directly training FilterGNN with $\mathrm{Att}_{\mathrm{linear}}$ from scratch does not cause the network to converge well. Note that neither $\mathrm{Att}_{\mathrm{std}}$ nor $\mathrm{Att}_{\mathrm{linear}}$ require additional learning parameters for the network. Therefore, we can adopt a two-step training approach: first, we use $\mathrm{Att}_{\mathrm{std}}$ for pretraining until convergence; second, we replace $\mathrm{Att}_{\mathrm{std}}$ with $\mathrm{Att}_{\mathrm{linear}}$ to finetune the network parameters. Thus, FilterGNN converges quickly without significant performance degradation. For a more detailed discussion, please refer to Section 4.3.

3.2.3 Top-K filter

For each GMF module, the score matrix $\mathcal{S} \in \mathbb{R}^{N,M}$ is defined as the dot product between the refined features:

$$\mathcal{S} = \widetilde{\mathcal{F}}_{\mathrm{A}} \otimes \widetilde{\mathcal{F}}_{\mathrm{B}}^{\mathrm{T}} \tag{6}$$

We expand $\mathcal{S}$ to $\widetilde{\mathcal{S}} \in \mathbb{R}^{N+1,M+1}$ by adding an additional learnable dustbin dimension for unmatched keypoint detection called SuperGlue [8]. Then, we adopt the dual-softmax [40] operator to produce an optimized matching confidence matrix $\mathcal{C}$ as Eq. (7):

$$\mathcal{C} = \log \mathrm{Softmax}(\widetilde{\mathcal{S}}) + \log \mathrm{Softmax}(\widetilde{\mathcal{S}}^{\mathrm{T}})^{\mathrm{T}} \tag{7}$$

Finally, we define the matching probability of each keypoint as the row (column)-wise maximum value of $\mathcal{C}$ (excluding the dustbin dimension), and filter out the lowest k keypoints. k is set to γN. Note that the NN search of the last layer in Fig. 2 illustrates a similar process. This step no longer considers the dustbin and directly determines the predicted matches based on the score matrix. This process is efficient because numerous outliers are filtered out in the previous layers. As shown in Figs. 1 and 4, the outliers detected by GMF are highly related to the covisibility. The source image outputs different outlier detection results for the different target images.

3.3 Loss function

We adopt a multilevel weighted loss function for H GMF modules as Eq. (8):

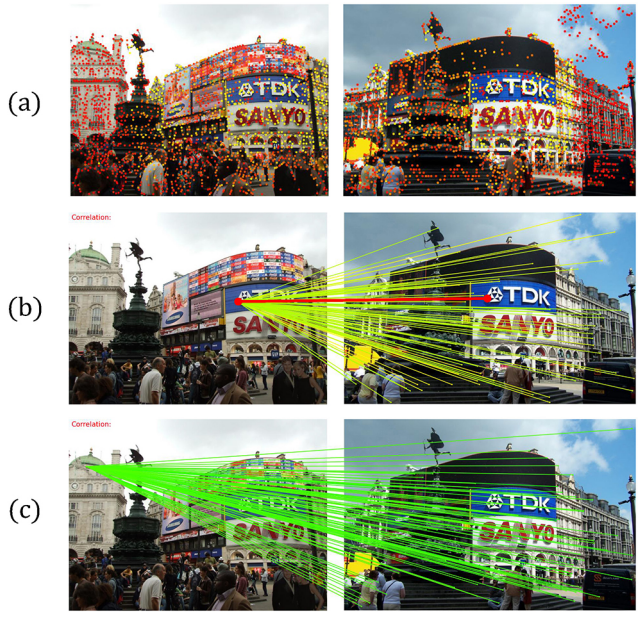


Fig. 4 (a) Visualization of matching probabilities: the color of the keypoints changes from yellow to red as the probability changes from high to low. (b) Visualization of the matching confidence vector of a matchable keypoint. (c) Visualization of the matching confidence vector of an unmatchable keypoint. The color of the lines changes from red to green indicates the confidence changing from high to low.

$$\mathcal{L} = \sum_h w_h \mathcal{L}_h, \quad w_h = 1 - \gamma h \tag{8}$$

to achieve a fast convergence and ensure the stability outlier filtering. γ represents the rejection ratio of each layer, which was set to 0.1. The matching loss L_h of the hth GMF module is defined as

$$\mathcal{L}_h = -\frac{1}{|\mathcal{M}|}\sum_{(i,j)\in\mathcal{M}} \mathcal{C}_{ij} - \frac{1}{|\mathcal{U}_\mathrm{A}|}\sum_{i\in\mathcal{U}_\mathrm{A}} \mathcal{C}_{i,m+1} - \frac{1}{|\mathcal{U}_\mathrm{B}|}\sum_{j\in\mathcal{U}_\mathrm{B}} \mathcal{C}_{n+1,j} \tag{9}$$

where $\mathcal{M}$ denotes the ground truth matching set, and $(\mathcal{U}_\mathrm{A},\mathcal{U}_\mathrm{B})$, the unmatchable keypoint sets. Combined with the definition of $\mathcal{C}$ in Eq. (7), we hope that the matching confidence vector of any matchable keypoint is as similar as possible to a one-hot vector (as shown in Fig. 4(b)). That is, the cosine similarity between unmatched keypoints should be as low as possible. However, the dimensions of the features are limited, and, as the input size increases, the angle between nonmatching points is reduced, increasing the possibility of mismatching. Note that the proposed outlier filtering mechanism can provide a relatively wide feature distribution space for deeper layers. Therefore, the validity of FilterGNN is theoretically guaranteed.

4 Experiments and discussions

4.1 Implementation details

Training dataset. FilterGNN was trained using the MegaDepth dataset [42], which contains 195 outdoor scenes with reconstructed camera poses and depth. We adopted the same training/validation split of 153/36 as reported in Ref. [8].

Visual descriptor. We used ASLfeat [31], a robust indoor/outdoor visual descriptor with 128 dimensions throughout all experiments. We extracted a maximum of 2048 keypoints for each image and randomly selected 1024 keypoints for data augmentation during training.

Architecture details. All feature representations $(\mathcal{Q},\mathcal{K},\mathcal{V},\mathcal{D},\mathcal{F})$ share the same 128 dimensions as ASLfeat. H and L were set to 3. The mlp_pe channels were set to (3, 64, 128, 256, 128), and those of mlp_ff were set to (128, 512, 128). Each linear layer (excluding the last layer) in both MLPs was followed by batch normalization and a ReLU layer. All attention layers mentioned were implemented with four-head multi-head attention. Our model was optimized using Adam, with an initial learning rate of 1×10^{-4} for the first 10 epochs, followed by an exponential decay of 0.9 for 20 epochs.

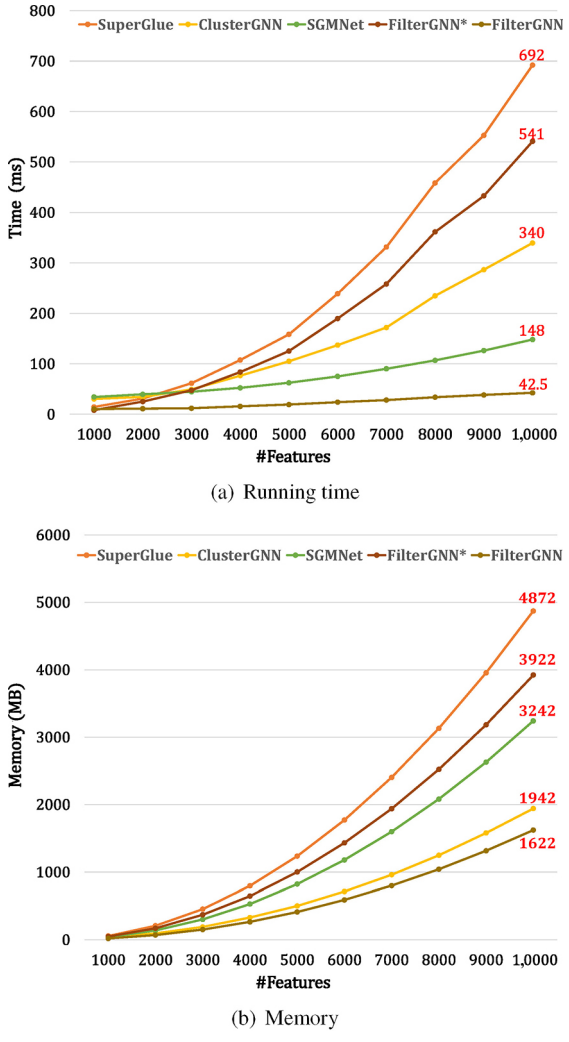
4.2 Results

We evaluated the efficiency and performance of our method by comparing it with SOTA methods, including SuperGlue [8], SGMNet [9], and ClusterGNN [10], on various computer vision tasks. ASLfeatV2 [31] is specified as the input image visual descriptor and currently one of the best descriptors applicable to both indoor and outdoor scenes.

4.2.1 Efficiency

All experiments were performed on the same NVIDIA GeForce RTX 3090 GPU. To clearly demonstrate the effect of the cascaded filter mechanism in FilterGNN, we adopted FilterGNN* to represent the corresponding results with standard attention for comparison.

First, we compared the running time and memory usage of the inference phase on a single GPU for the basic feature-matching task. Most statistics were averaged using the same batch size (four by default). For SuperGlue with a 10,000 input size, the batch size was set to three to avoid running out of memory. In Fig. 5, we report the running time and memory



(a) Running time

(b) Memory

Fig. 5 (a) Running time and (b) GPU memory consumption with respect to the number of input keypoints for different methods.

consumption for different numbers of input keypoints (ranging from 1000 to 10,000). The time complexity of our FilterGNN is linear with the input feature size, and our method significantly improved in both time and memory consumption compared with SOTA methods. In particular, when the number of input keypoints is 10,000, our method requires only 6% of the time cost and 33.3% of the memory consumption of that of SuperGlue. In the following reports, we used 2000, 4000, and 6000 input points for different experiments.

4.2.2 Pose estimation

Camera pose estimation is one of the most important applications of local feature matching, for which RANSAC postprocessing is typically adopted to filter correspondences. Following SuperGlue, we evaluated the accuracy of the location estimation on the **YFCC100M** [14] benchmark, which contains 4000 test image pairs with ground truth relative poses and known camera intrinsics. In addition to SuperGlue, SGMNet, and ClusterGNN, we used NN search as the baseline to evaluate the performance of the raw input ASLfeatV2 descriptors. In this experiment, the number of input keypoints was set to 2048. Table 1 reports the success rates according to the area under curve (AUC) metric [1, 43, 44] with three different thresholds (i.e., 5°, 10°, and 20°), which combine both the rotation and translation error. In addition, we report the precision (P) of the matches and ratio between the number of matches and input size (MS). All indicators are better for larger numbers. FilterGNN* significantly outperforms current SOTA methods, indicating that the cascaded filter mechanism does not exclude unmatchable keypoints and increases the number of matches, thereby contributing to a more accurate pose estimation. FilterGNN uses linear attention to achieve a competitive performance compared with SOTA methods, but with lower time and memory consumptions, as demonstrated. Note that the high prediction accuracy of SuperGlue [8] was based on fewer predicted matches. We can infer that SuperGlue [8] is conservative, whereas FilterGNN is more aggressive and capable of solving fine-grained problems. Figure 6 presents the qualitative results.

4.2.3 Visual localization

Visual localization is another important application in local image feature matching. A typical pipeline comprises image retrieval, image matching, and a perspective n-point pose solver. Both the number and accuracy of matches affect the localization precision. We integrated FilterGNN into the official HLoc [4] pipeline and conducted experiments on the long-term visual localization benchmark [45]. In particular, we

Table 1 Pose estimation on the YFCC100M benchmark. The best result is in bold, and second-best is underlined

Matcher	AUC (↑)			P (↑)	MS (↑)
	5°	10°	20°		
NN	27.95	45.20	61.17	54.29	14.29
SuperGlue	39.92	59.93	<u>76.03</u>	**99.16**	15.55
SGMNet	32.22	52.53	70.16	—	—
ClusterGNN	35.31	56.13	73.56	—	—
FilterGNN	<u>40.91</u>	<u>60.15</u>	75.47	91.26	<u>23.92</u>
FilterGNN*	**44.25**	**63.81**	**78.65**	<u>94.46</u>	**32.59**

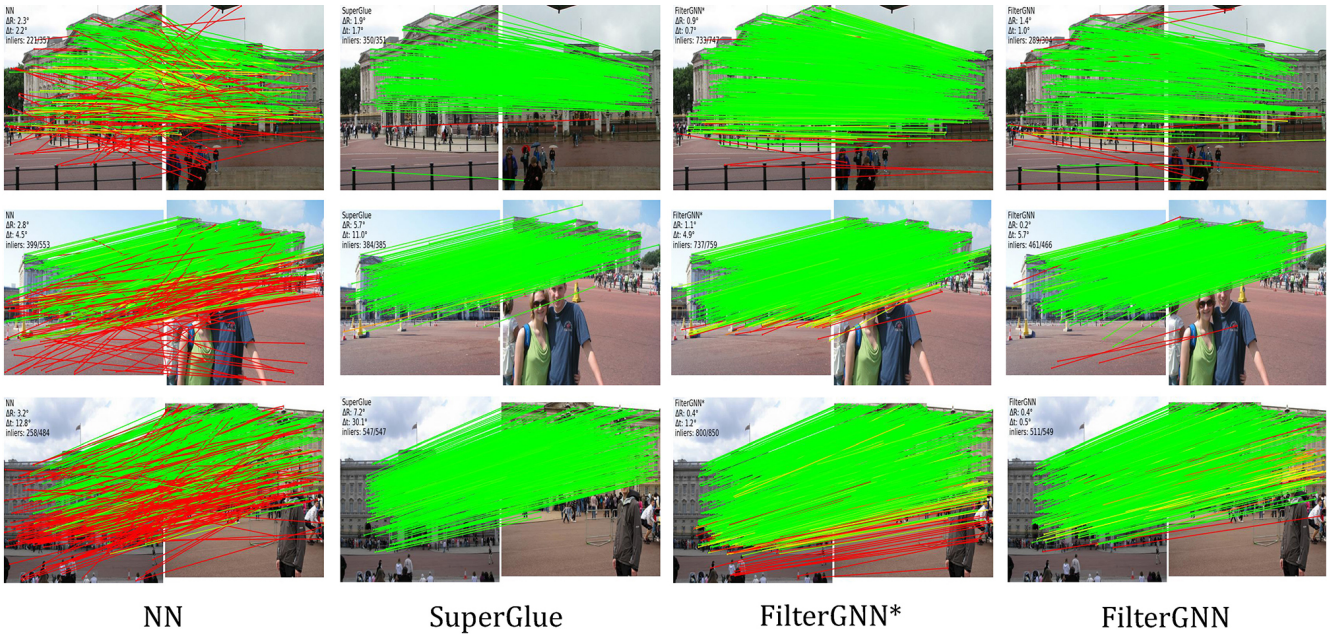


Fig. 6 Qualitative examples of YFCC100M. The red and green lines indicate the outliers and inliers, respectively. The rotation error, translation error, and number of inliers/matches are shown in the upper left corner of each image.

selected two representative datasets: the Aachen day–night dataset [16, 17] (outdoor) and InLoc dataset [18] (indoor).

The Aachen day–night dataset contains 922 query and 824/98 daytime/nighttime images. All images were taken around the same street in Aachen, where the sparseness of views and day–night variation were the main challenges.

The InLoc dataset contains 329 query and 9972 database images. The main challenges include complex lighting conditions and common textureless objects (floors, ceilings, and walls).

The number of input keypoint was set to 4096. We reported the percentage of correctly localized queries under different thresholds (referring to the leaderboard of the long-term visual localization benchmark [45]). Tables 2 and 3 list the results for the Aachen day–night and InLoc datasets, respectively.

Table 2 Outdoor localization results on Aachen day-night benchmark (v1.0). The best result is in bold

Method	Day	Night
	(0.25 m, 2°) / (0.5 m, 5°) / (1.0 m, 10°)	
NN	82.3 / 89.2 / 92.7	67.3 / 79.6 / 85.7
Superglue	87.9 / 95.4 / 98.3	81.6 / 91.8 / 99.0
ClusterGNN	88.6 / **95.5** / 98.4	**85.7** / **93.9** / 99.0
FilterGNN*	**89.2** / 95.4 / 98.5	**85.7** / 92.9 / **100.0**
FilterGNN	88.7 / 95.4 / **98.7**	84.7 / 92.9 / **100.0**

Table 3 Indoor localization results on InLoc dataset. The best result is in bold

Method	DUC1	DUC2
	(0.25 m, 10°) / (0.5 m, 10°) / (1.0 m, 10°)	
NN	40.4 / 58.1 / 67.7	35.9 / 52.7 / 60.3
SuperGlue	51.5 / 66.7 / 75.8	53.4 / **76.3** / **84.0**
ClusterGNN	52.5 / 68.7 / 76.8	55.0 / 76.0 / 82.4
FilterGNN*	**55.6** / **69.7** / **78.8**	**59.5** / 75.6 / 77.1
FilterGNN	52.5 / 67.7 / 77.8	58.0 / 77.1 / 82.4

Our method achieved results comparable to those of other SOTA methods.

4.2.4 Sparse reconstruction

To demonstrate the robustness of FilterGNN more intuitively, we integrated it into the COLMAP [1] pipeline for sparse 3D reconstruction. We extracted 6000 keypoints per image for this task. **THU-MVS dataset** [46] contains multi-view images with 3D ground truths consisting of two cases: (a) a cat model with 108 views and (b) a dog model with 72 views. Typically, animal images have weakly textured body surfaces that challenge local feature matching. As shown in Fig. 7, FilterGNN performed the best in terms of both the reconstruction density and accuracy. Compared with SuperGlue, our method increases the reconstruction density by 72% on average and reduces the reconstruction error by 35%.

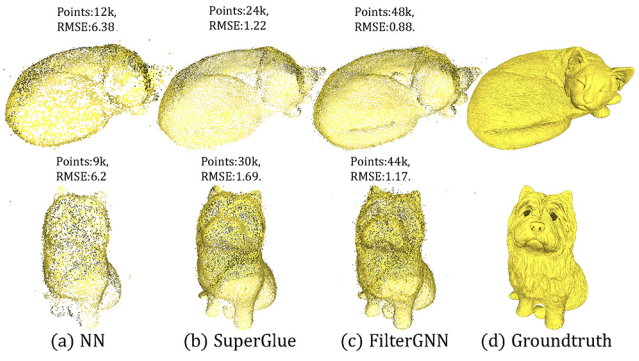


Fig. 7 Visualization of sparse reconstruction on the THU-MVS dataset [46]. The size of the reconstruction point cloud and reconstruction error are displayed above the corresponding model. RMSE refers to the root-mean-square error.

4.3 Ablation study and discussion

The ablation study addressed two aspects: the submodule composition and structure. First, we report the effects of the post-norm method, attention method, and optimal matching function by comparing the matching accuracy (precision and recall) on the validation part of the MegaDepth dataset [42] (with all training epochs set to 20). The post-norm has two norm options: layer and instance. As the optimal matching functions (Opt.), as reported in previous works [10, 11], both Sinkhorn [47] and dual-softmax [40] work well in most cases. For the attention computation, we tested the four most representative methods:

- **Standard** scale product attention implemented in transformer [7];
- **Linear attention** (LA) [25] with both kernel functions (ϕ, φ) set to $1 + \text{elu}$;
- **Performer** [23] with an additional low-dimension linear projection and kernel functions (ϕ, φ) set to $(\text{softmax}, \text{exponential})$, respectively; and
- **Efficient attention** (EA) [39], which was adopted in our work, with both kernel functions (ϕ, φ) set to softmax.

Table 4 presents the comparison results. The following conclusions can be drawn. (i) The instance norm always has a positive effect, whereas the layer norm leads to a negative effect. (ii) All linear attention methods, except efficient attention [39], cause a significant drop, which significantly differs from the reported evaluation of vision tasks or natural language processing tasks. (iii) Pretraining with standard attention significantly improves the performance of efficient attention [39] but has no significant effect on other methods. Directly adopting the traditional linear attention method reduces the accuracy rate by 30%, and the pretrained strategy reduces the performance to under 8%. The two-stage pretraining strategy results in an improvement of approximately 10%. Note that, in some cases, dual-softmax causes abnormal gradients. We speculate that this may be caused by the accumulation of data variance in the residual network, which can be effectively avoided using an instance norm.

Table 4 Comparison different post-norms, attention methods, and optimal matching functions. LN and IN indicate a layer or instance norm, respectively. sh and ds indicate Sinkhorn or dual-softmax, respectively; "any" means that the choice has no appreciable changes on the results

Attention	Norm	Pretrain	Opt.	P/R
Standard	—	—	sh	78.1/88.5
Standard	—	—	ds	81.6/88.2
Standard	LN	—	sh	67.0/50.0
Standard	LN	—	ds	NAN
Standard	IN	—	any	86.8/87.8
Performer [23]	—	—	any	54.3/59.6
Performer [23]	IN	any	any	58.4/62.9
LA [25]	—	—	sh	55.9/56.2
LA [25]	—	—	ds	NAN
LA [25]	IN	any	any	59.3/57.6
EA [39]	—	—	any	62.6/60.9
EA [39]	IN	—	any	70.2/74.5
EA [39]	—	✓	any	78.3/79.0
EA [39]	IN	✓	any	80.3/80.1

We report the effect of the network structure, including the number of GMFs H, filtering ratio of the GMFs γ, and number of attention blocks L in each attention GNN module of the GMF. As shown in Table 5, increasing the value of H or L from the default contributes to limited improvement. An excessively large γ reduces the recall, which is unsuitable for other postprocessing tasks.

Table 5 Ablation of FilterGNN structure. The bold row presents the default settings

Method	H	L	γ	P/R
FilterGNN	3	3	0.1	78.0/81.5
	3	**3**	**0.2**	**80.3/80.1**
	3	3	0.3	92.0/65.7
	3	2	0.2	74.2/78.3
	3	4	0.2	80.5/80.2
	2	3	0.2	77.9/75.7
	4	3	0.2	85.2/72.3

5 Conclusions

This study presents FilterGNN, which is an efficient and novel approach for local image feature matching. In observing the high-sparsity property of long-term image feature matching, we designed hierarchical filter blocks to remove unmatchable keypoints in a cascading manner. This instant outlier removal mechanism adjusts the network focus to the keypoints within covisible regions, which is beneficial for solving fine-grained problems. The results verified that FilterGNN can substantially increase the number of predicted matches, which is crucial for both accurate visual localization and high-quality sparse reconstruction. Moreover, we strictly reduce the time complexity from $O(N^2)$ to $O(N)$ by introducing a two-stage pretraining-finetuning strategy without obvious performance degradation, which many tasks have validated. As FilterGNN can reach a running speed of 50 Hz with an input of 10,000 keypoints, we plan to apply it to more complex scenes in the future, such as 3D point clouds [48] and super-resolution images, which require more input feature points.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 62220106003) and Tsinghua–Tencent Joint Laboratory for Internet Innovation Technology.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Schonberger, J. L.; Frahm, J. M. Structure-from-motion revisited. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4104–4113, 2016.

[2] Mur-Artal, R.; Montiel, J. M. M.; Tardós, J. D. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics* Vol. 31, No. 5, 1147–1163, 2015.

[3] Huang, J.; Yang, S.; Zhao, Z.; Lai, Y. K.; Hu, S. M. ClusterSLAM: A SLAM backend for simultaneous rigid body clustering and motion estimation. *Computational Visual Media* Vol. 7, No. 1, 87–101, 2021.

[4] Sarlin, P. E.; Cadena, C.; Siegwart, R.; Dymczyk, M. From coarse to fine: Robust hierarchical localization at large scale. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12716–12725, 2019.

[5] Dusmanu, M.; Rocco, I.; Pajdla, T.; Pollefeys, M.; Sivic, J.; Torii, A.; Sattler, T. D2-net: A trainable CNN for joint detection and description of local features. *arXiv preprint* arXiv:1905.03561, 2019.

[6] DeTone, D.; Malisiewicz, T.; Rabinovich, A. SuperPoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 224–236, 2018.

[7] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In: Proceedings of the 31st Conference on Neural Information Processing Systems, 5998–6008, 2017.

[8] Sarlin, P. E.; DeTone, D.; Malisiewicz, T.; Rabinovich, A. SuperGlue: Learning feature matching with graph neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4938–4947, 2020.

[9] Chen, H.; Luo, Z.; Zhang, J.; Zhou, L.; Bai, X.; Hu, Z.; Tai, C. L.; Quan, L. Learning to match features with seeded graph matching network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 6301–6310, 2021.

[10] Shi, Y.; Cai, J. X.; Shavit, Y.; Mu, T. J.; Feng, W.; Zhang, K. ClusterGNN: Cluster-based coarse-to-fine graph neural network for efficient feature matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12517–12526, 2022.

[11] Sun, J.; Shen, Z.; Wang, Y.; Bao, H.; Zhou, X. LoFTR: Detector-free local feature matching with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8922–8931, 2021.

[12] Suwanwimolkul, S.; Komorita, S. Efficient linear attention for fast and accurate keypoint matching. In: Proceedings of the International Conference on Multimedia Retrieval, 330–341, 2022.

[13] Guo, M. H.; Xu, T. X.; Liu, J. J.; Liu, Z. N.; Jiang, P. T.; Mu, T. J.; Zhang, S. H.; Martin, R. R.; Cheng, M. M.; Hu, S. M. Attention mechanisms in computer vision: A survey. *Computational Visual Media* Vol. 8, No. 3, 331–368, 2022.

[14] Thomee, B.; Elizalde, B.; Shamma, D. A.; Ni, K.;

Friedland, G.; Poland, D.; Borth, D.; Li, A. L. J. YFCC100M: The new data in multimedia research. *Communications of the ACM* Vol. 59, No. 2, 64–73, 2016.

[15] Balntas, V.; Lenc, K.; Vedaldi, A.; Mikolajczyk, K. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5173–5182, 2017.

[16] Sattler, T.; Weyand, T.; Leibe, B.; Kobbelt, L. Image retrieval for image-based localization revisited. In: Proceedings of the British Machine Vision Conference, 2012.

[17] Zhang, Z.; Sattler, T.; Scaramuzza, D. Reference pose generation for long-term visual localization *via* learned features and view synthesis. *International Journal of Computer Vision* Vol. 129, No. 4, 821–844, 2021.

[18] Taira, H.; Okutomi, M.; Sattler, T.; Cimpoi, M.; Pollefeys, M.; Sivic, J.; Pajdla, T.; Torii, A. InLoc: Indoor visual localization with dense matching and view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7199–7209, 2018.

[19] Qiu, J.; Ma, H.; Levy, O.; Yih, S. W. T.; Wang, S.; Tang, J. Blockwise self-attention for long document understanding. *arXiv preprint* arXiv:1911.02972, 2019.

[20] Parmar, N.; Vaswani, A.; Uszkoreit, J.; Kaiser, Ł.; Shazeer, N.; Ku, A.; Tran, D. Image transformer. *arXiv preprint* arXiv:1802.05751, 2018.

[21] Kitaev, N.; Kaiser, Ł.; Levskaya, A. Reformer: The efficient transformer. *arXiv preprint* arXiv:2001.04451, 2020.

[22] Wang, S.; Li, B. Z.; Khabsa, M.; Fang, H.; Ma, H.; Kolodiazhnyi, K. Linformer: Self-attention with linear complexity. *arXiv preprint* arXiv:2006.04768, 2020.

[23] Choromanski, K.; Likhosherstov, V.; Dohan. D.; Song, X.; Gane, A.; Sarlos, T.; Hawkins, P.; Davis, J.; Mohiuddin, A.; Kaiser, L.; et al. Rethinking attention with performers. In: Proceedings of the International Conference on Learning Representations, 2021.

[24] Guo, M. H.; Liu, Z. N.; Mu, T. J.; Hu, S. M. Beyond self-attention: External attention using two linear layers for visual tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 45, No. 5, 5436–5447, 2023.

[25] Katharopoulos, A.; Vyas, A.; Pappas, N.; Fleuret, F. Transformers are RNNs: Fast autoregressive transformers with linear attention. *arXiv preprint* arXiv:2006.16236, 2020.

[26] Gu, Y.; Qin, X.; Peng, Y.; Li, L. Content-augmented feature pyramid network with light linear spatial transformers for object detection. *IET Image Processing* Vol. 16, No. 13, 3567–3578, 2022.

[27] Ulyanov, D.; Vedaldi, A.; Lempitsky, V. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint* arXiv:1607.08022, 2016.

[28] Ba, J. L.; Kiros, J. R.; Hinton, G. E. Layer normalization. *arXiv preprint* arXiv:1607.06450, 2016.

[29] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* Vol. 60, No. 2, 91–110, 2004.

[30] Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In: Proceedings of the International Conference on Computer Vision, 2564–2571, 2011.

[31] Luo, Z.; Zhou, L.; Bai, X.; Chen, H.; Zhang, J.; Yao, Y.; Li, S.; Fang, T.; Quan, L. ASLFeat: Learning local features of accurate shape and localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6589–6598, 2020.

[32] Yi, K. M.; Trulls, E.; Lepetit, V.; Fua, P. LIFT: Learned invariant feature transform. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9910*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 467–483, 2016.

[33] Mishkin, D.; Radenović, F.; Matas, J. Repeatability is not enough: Learning affine regions via discriminability. In: Proceedings of the European Conference on Computer Vision, 284–300, 2018.

[34] Devlin, J.; Chang, M. W.; Lee, K.; Toutanova, K.; Hulburd, E.; Liu, D.; Wang, M.; Catlin, A. G.; Lei, M.; Zhang, J.; et al. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the NAACL-HLT, 4171–4186, 2018.

[35] Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint* arXiv:2010.11929, 2020.

[36] Child, R.; Gray, S.; Radford, A.; Sutskever, I. Generating long sequences with sparse transformers. *arXiv preprint* arXiv:1904.10509, 2019.

[37] Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 10012–10022, 2021.

[38] Roy, A.; Saffar, M.; Vaswani, A.; Grangier, D. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics* Vol. 9, 53–68, 2021.

[39] Shen, Z.; Zhang, M.; Zhao, H.; Yi, S.; Li, H. Efficient attention: Attention with linear complexities. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, 3531–3539, 2021.

[40] Rocco, I.; Cimpoi, M.; Arandjelović, R.; Torii, A.; Pajdla, T.; Sivic, J. Neighbourhood consensus networks. In: Proceedings of the 32nd Conference on Neural Information Processing Systems, 1658–1669, 2021.

[41] Liu, Z.; Hu, H.; Lin, Y.; Yao, Z.; Xie, Z.; Wei, Y.; Ning, J.; Cao, Y.; Zhang, Z.; Dong, L.; et al. Swin transformer V2: Scaling up capacity and resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12009–12019, 2022.

[42] Li, Z.; Snavely, N. MegaDepth: Learning single-view depth prediction from Internet photos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2041–2050, 2018.

[43] Ono, Y.; Trulls, E.; Fua, P.; Yi, K. M. LF-Net: Learning local features from images. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 6237–6247, 2018.

[44] Schönberger, J. L.; Zheng, E.; Frahm, J. M.; Pollefeys, M. Pixelwise view selection for unstructured multi-view stereo. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9907*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 501–518, 2016.

[45] Toft, C.; Maddern, W.; Torii, A.; Hammarstrand, L.; Stenborg, E.; Safari, D.; Okutomi, M.; Pollefeys, M.; Sivic, J.; Pajdla, T.; et al. Long-term visual localization revisited. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 44, No. 4, 2074–2088, 2022.

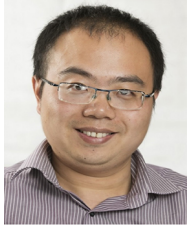
[46] Sakai, S.; Ito, K.; Aoki, T.; Watanabe, T.; Unten, H. Phase-based window matching with geometric correction for multi-view stereo. *IEICE Transactions on Information and Systems* Vol. E98.D, No. 10, 1818–1828, 2015.

[47] Sinkhorn, R.; Knopp, P. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics* Vol. 21, No. 2, 343–348, 1967.

[48] Guo, J.; Wang, H.; Cheng, Z.; Zhang, X.; Yan, D. M. Learning local shape descriptors for computing non-rigid dense correspondence. *Computational Visual Media* Vol. 6, No. 1, 95–112, 2020.

Jun-Xiong Cai is currently a researcher at Huawei; previously, he was a postdoctoral researcher at Tsinghua University. He received his Ph.D. degree in computer science and technology in 2020 from Tsinghua University. His research interests include computer graphics, computer vision, and 3D geometric processing.

Tai-Jiang Mu is an assistant researcher with the Department of Computer Science and Technology at Tsinghua University. He received his bachelor and Ph.D. degrees in computer science and technology from Tsinghua University in 2011 and 2016, respectively. His research interests include computer graphics, visual-media learning, 3D reconstruction, and 3D understanding.



Yu-Kun Lai received his bachelor and Ph.D. degrees in computer science from Tsinghua University in 2003 and 2008, respectively. He is currently a professor at the School of Computer Science and Informatics, Cardiff University, UK. His research interests include computer graphics, geometric and image processing, and computer vision. He is on the editorial board of *The Visual Computer*. He is a member of IEEE.