Collaborative Feature Matching with Progressive Correspondence Learning

Xin Liu[1*], **Yanbing Han**[1*], **Rong Qin**[1], **Bing Wang**[4], **Jufeng Yang**[1, 2, 3†]
[1]VCIP & TMCC & DISSec, College of Computer Science, Nankai University
[2]Pengcheng Laboratory
[3] Nankai International Advanced Research Institute (SHENZHEN· FUTIAN)
[4]Faculty of Engineering, Hong Kong Polytechnic University
xinliu_0209@163.com, {2112434, qinrong_nk}@mail.nankai.edu.cn, bingwang@polyu.edu.hk, yangjufeng@nankai.edu.cn

Abstract

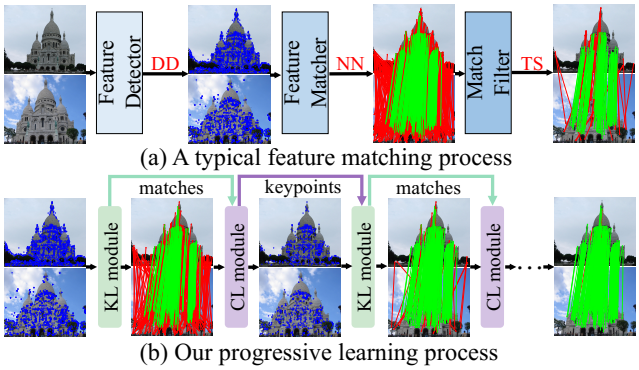
Accurate feature matching between image pairs is fundamental for various computer vision applications. In detector-base process, the feature matcher aims to find the optimal feature correspondences, and the match filter is used for further removing mismatches. However, their connection is rarely exploited since they are usually treated as two separate issues in previous method, which may lead to suboptimal results. In this paper, we propose an end-to-end collaborative feature matching (CFM) method, which contains a keypoint learning (KL) module and a correspondence learning (CL) module, to bridge the gap between two types of works. The former improves the discrimination of keypoints, and provides high-quality dynamic matches for CL module. The latter further captures the rich context of matches, and gives effective feedback to KL module. These two modules can reinforce each other in a progressive manner. Besides, we develop an efficient version of CFM, named ECFM, using an adaptive sampling strategy to avoid the negative influence of uninformative keypoints. Experimental results indicate that both methods outperform the state-of-the-art competitors in the tasks of relative pose estimation and visual localization.

Code — https://github.com/xinliu29/CFM

1 Introduction

Detector-base feature matching aims at estimating correct feature point-to-point correspondences (matches) to recover the geometric relationship of image pairs, which is indispensable for a variety of computer vision applications (Ma et al. 2021; Liu et al. 2022; Xiao et al. 2024).

As shown in Fig. 1 (a), feature keypoints encoding the local visual appearance can first be obtained through traditional or learning-based feature detectors (Lowe 2004; DeTone, Malisiewicz, and Rabinovich 2018). Then, matches are established by the nearest neighbor (NN) search on feature descriptors with some heuristic tricks. However, these matches inevitably contain a large number of mismatches ($i.e.$, outliers) due to the ambiguity and limited representation ability of keypoints, especially for some challenging matching scenes, such as large viewpoint changes, occlusions, blurs, and repetitive structures.



(a) A typical feature matching process

(b) Our progressive learning process

Figure 1: (a) A typical detector-base feature matching process containing three main steps. (b) The progressive learning process of our CFM. With the collaborative effect, it can obtain accurate matches (more green lines and fewer red lines). **DD**: detection and description. **NN**: nearest neighbor search. **TS**: threshold selection.

To alleviate this problem, a lot of learning-based feature matchers (Sarlin et al. 2020; Chen et al. 2021; Shi et al. 2022) conduct intra/inter-graph communication of keypoints for augmenting the representation ability of feature descriptors. Then, they adopt the Sinkhorn algorithm (Cuturi 2013) to reject non-matchable keypoints with low matching confidence. Nevertheless, the paired relationship of keypoints, $e.g.$, correct matches (inliers) typically adhere to consistent constraints (such as lengths, angles, and motion), is rarely further explored in previous methods since they primarily focus on individual keypoints. At the same time, some match filters (Yi et al. 2018; Cavalli et al. 2020; Dai et al. 2022; Dai, Du, and Tang 2024) further screen the predicted matches by employing the geometric property of matches or neural networks. For example, PointCN (Yi et al. 2018) solves this problem in a correspondence classification manner based on multi-layer perceptrons (MLPs) to remove possible outliers. However, match filters have to tolerate the quality of input matches that have been determined in advance, restricting their performance ceiling (Liu et al. 2024a).

Moreover, in existing works, feature matcher and match filter are addressed as separate issues (Ma et al. 2021). A naive assumption is that combining them in a sequential

*These authors contributed equally.
†Corresponding author.

manner can yield more accurate results. However, this strategy is hardly effective (see ablation studies). We suggest that this might be the lack of information interaction and joint optimization, leading to incompatibility. Meanwhile, the existing match filter, as a post-processing step, fails to provide feedback to the feature matcher for improving accuracy.

To further explore their potential, we present an end-to-end method, called collaborative feature matching (CFM), to alleviate this incompatibility between the two types of methods. It consists of two components: keypoint learning (KL) module and correspondence learning (CL) module. The embeddings learned from both modules can be used with each other for mutual enhancement and joint optimization. Specifically, in KL module, self attention and cross attention aggregate the spatial and visual contexts based on the intra-graph and inter-graph, to enhance the representation ability of descriptors as (Sarlin et al. 2020). With descriptors becoming more discriminative, high-quality matches can be provided, and the enhanced descriptors are also exploited as feature embedding to enrich the inputs of CL module. In CL module, we design a local consensus block based on the spatial and feature similarity to effectively aggregate rich contexts of matches, and calculate the inlier scores. To improve the learning of KL module, inlier confidence learned from CL module serves as compensation to further boost the representation of keypoints, which brings significant gains. Meanwhile, we propose an adaptive sampling strategy via the learned inlier confidence to avoid the negative influence of uninformative keypoints that are usually not in the overlapping region. With this collaborative effect, these two modules can enhance each other in a progressive manner as illustrated in Fig. 1 (b) to improve the quality of feature matches. Furthermore, an efficient version of CFM, called ECFM, is developed using the adaptive sampling strategy.

Our contributions are threefold: (1) We present a CFM that integrates both KL module and CL module, which can progressively reinforce each other with well-designed structures, to bridge the gap between the two types of works. (2) KL module is responsible for boosting the representation of keypoints and providing high-quality dynamic matches. While CL module further aggregates rich context of these matches and provides reliable keypoint feedback. (3) Experimental results demonstrate the effectiveness and rationality of collaborative learning between the two features.

2 Related Work

Feature matching is typically addressed through detector-free or detector-based methods. The former (Edstedt et al. 2024; Wang et al. 2024) obtain dense pixel-wise matches by directly handling image pairs, which demands expensive computational resources due to the immense volume of pixels. In contrast, the latter (Lowe 2004; Lee et al. 2023) utilizes fewer distinctive keypoints to construct sparse point-wise matches. It typically contains the following steps: 1) feature detector, 2) feature matcher, 3) match filter (Ma et al. 2021). Recent researchers utilize neural networks to improve the different steps of detector-based feature matching.

Feature detector. SIFT (Lowe 2004) and ORB (Rublee et al. 2011) are arguably the most successful hand-crafted feature detectors widely adopted in many computer vision tasks. Subsequently, many approaches adopt CNNs (He et al. 2016) or Transformer (Vaswani et al. 2017) to obtain distinctive and reliable positions and descriptors for local keypoints. Notably, LIFT (Yi et al. 2016) is the first successful learning-based local feature detector. SuperPoint (DeTone, Malisiewicz, and Rabinovich 2018) proposes a self-supervised training method by homographic adaptation. Along this line, subsequent learning-based local feature detectors (Potje et al. 2023; He et al. 2023) adopt different strategies to detect distinctive keypoints. However, the representation and repeatability of keypoints are difficult to guarantee since they only operate on the single image, leading to unsatisfactory matching results (Ma et al. 2021).

Feature matcher. Recently, the graph neural network (GNN) (Wu et al. 2022) has gained widespread attention to boost matching accuracy. SuperGlue (Sarlin et al. 2020) accepts two keypoint sets as inputs and constructs their communication with attentional GNN. Since the priors can be learned with a data-driven approach, it achieves impressive performance. Subsequent variants (Chen et al. 2021; Pautrat et al. 2023; Jiang et al. 2024) design various network paradigms to further improve the performance. These works aim to enhance the representation of descriptors through the interaction of two keypoint sets. However, the paired relationship of keypoints is rarely exploited in these efforts, which is vital to distinguish matches.

Match filter. Concurrently, some pioneer works (Yi et al. 2018; Ranftl and Koltun 2018) take the feature matches as inputs and learn to screen these matches. PointCN (Yi et al. 2018) achieves this goal based on a classification task and a regression task. It implements a permutation-equivariant architecture based on MLPs to handle unordered and irregular correspondences. Subsequent methods (Zhang et al. 2019; Zhao et al. 2021; Ye et al. 2023; Liu et al. 2024b; Dai et al. 2024; Liu, Li, and Zhao 2025) improve the network performance by designing various network structures. These methods have shown the advantage of correspondence learning and could serve as an effective method to remove possible outliers. However, their performance heavily depends on the quality of initial matches, which limits their potential.

3 Methodology

3.1 Problem Formulation

Given a pair of images $(\mathrm{I_x}, \mathrm{I_y})$, feature matching aims to establish accurate point-to-point correspondences. Specifically, we can first utilize existing feature detectors ($e.g.$, SIFT and SuperPoint) to obtain initial keypoints $\mathbf{X}^{(0)} = \left\{\mathbf{x}_1^{(0)}, ..., \mathbf{x}_{n_1}^{(0)}\right\}$ and $\mathbf{Y}^{(0)} = \left\{\mathbf{y}_1^{(0)}, ..., \mathbf{y}_{n_2}^{(0)}\right\}$, where n_1 and n_2 are the number of keypoints. Each keypoint $\mathbf{x}_i^{(0)} = (\mathbf{k}_i, \mathbf{d}_i^{(0)})$ consists of keypoint information $\mathbf{k}_i \in \mathbb{R}^3$ and initial local descriptor $\mathbf{d}_i^{(0)} \in \mathbb{R}^d$. d is the dimension of descriptors. Each $\mathbf{k}_i$ contains normalized coordinates and a detection score. Usually, matches can be produced by using NN search on descriptors. However, when the image pair faces large differences, the simple NN search may produce a large

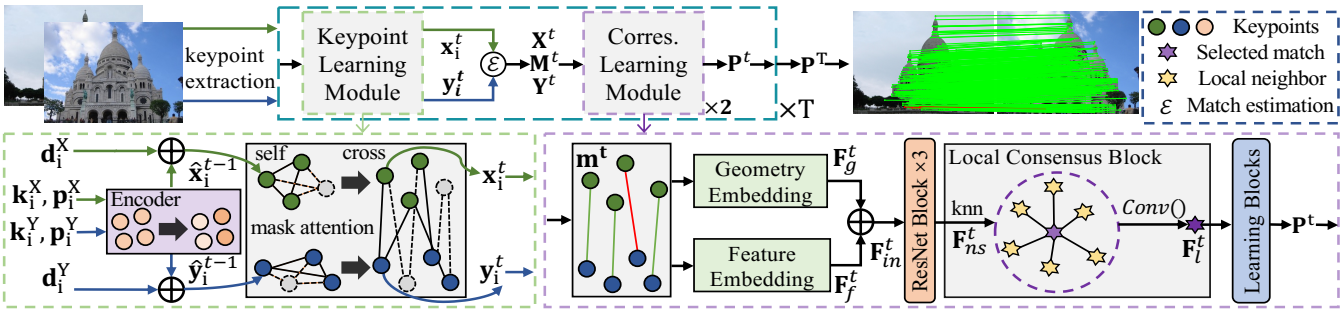


Figure 2: Overall pipeline of CFM. KL module provides high-quality matches. CL module gives reliable keypoint feedback. $\mathbf{x}_i$ and $\mathbf{y}_i$ denote feature keypoints containing descriptor $\mathbf{d}_i$, keypoint information $\mathbf{k}_i$ and inlier probability $\mathbf{p}_i$. $\mathbf{M}^t$ and $\mathbf{P}^t$ are estimated match set and inlier probability set. $\mathbf{m}^t$ is the input match. $\mathbf{F}_g^t$ and $\mathbf{F}_f^t$ are geometry embedding and feature embedding. $\mathbf{F}_{in}^t$ is the input feature map. $\mathbf{F}_{ns}^t$ is the neighbor search space. $\mathbf{F}_l^t$ is the local context feature. **Corres.**: Correspondence.

number of mismatches due to the ambiguity of descriptors. Therefore, we use CFM to amplify the representation of keypoints and further establish accurate matches. Finally, we can get an optimal match set $\mathbf{M} = \{\mathbf{m}_1, ..., \mathbf{m}_{n_3}\}$, which includes keypoint coordinates. n_3 is the number of matches.

3.2 Progressive Framework

The correlation between feature matcher and match filter is rarely explored in previous methods, which results in a lot of useful information being wasted. In this work, we take full advantage of two types of methods, where KL module and CL module can achieve iteratively reinforcing in a progressive framework as shown in Fig. 2.

Visual descriptors encode the essential information of keypoints ($e.g.$, color, texture, and intensity), which form the foundations for matching keypoints. However, these descriptors may have poor representation and repeatability since feature detectors only operate on the local region of single image. Therefore, KL module is used to improve the representation of visual descriptors based on the communication of keypoints. It takes the $\mathbf{X}^{(t-1)}$, $\mathbf{Y}^{(t-1)}$, keypoint feedback $\mathbf{P}^{(t-1)}$ (if available) as inputs. With the learning of KL module, we can obtain updated keypoints $\mathbf{X}^{(t)}$ and $\mathbf{Y}^{(t)}$ with stronger representation of descriptors. Then, we utilize match estimation to get a match set $\mathbf{M}^{(t)}$. In practice, $\mathbf{M}^{(t)}$ still contains a proportion of mismatches due to the limited local receptive field of keypoints. Thus, CL module aims to further remove them as much as possible based on the consistency of inliers. It takes the $\mathbf{M}^{(t)}$, $\mathbf{X}^{(t)}$ and $\mathbf{Y}^{(t)}$ as inputs. With the learning of CL module, we can get a corresponding inlier probability set $\mathbf{P}^{(t)}$ to determine refined matches as the output. The overall process can be represented as:

$$\mathbf{X}^{(t)}, \mathbf{Y}^{(t)} = \mathcal{K}(\mathbf{X}^{(t-1)}, \mathbf{Y}^{(t-1)}, \mathbf{P}^{(t-1)}), \tag{1}$$

$$\mathbf{M}^{(t)} = \mathcal{E}(\mathbf{X}^{(t)}, \mathbf{Y}^{(t)}), \tag{2}$$

$$\mathbf{P}^{(t)} = \mathcal{C}(\mathbf{M}^{(t)}, \mathbf{X}^{(t)}, \mathbf{Y}^{(t)}), \tag{3}$$

where $\mathcal{K}$, $\mathcal{C}$, and $\mathcal{E}$ represent KL module, CL module, and match estimation (NN search in this paper), respectively. This process is executed iteratively for reliable matches.

Meanwhile, the embeddings learned from both modules can collaborate with each other for mutual enhancement and joint optimization. Firstly, as the descriptors become more discriminative, KL module is able to provide dynamic and reliable input matches for CL module. Different from previous works that use only the keypoint coordinates, CL module combines both geometry embedding and feature embedding to enrich the input feature maps. Specifically, an MLP layer and a ResNet block (Yi et al. 2018) are adopted to process match coordinates to obtain geometry embedding $\mathbf{F}_g^{(t)} \in \mathbb{R}^{n \times d_m}$. We use the same operation on match descriptors to produce feature embedding $\mathbf{F}_f^{(t)} \in \mathbb{R}^{n \times d_m}$:

$$\mathbf{F}_g^{(t)}, \mathbf{F}_f^{(t)} = \mathrm{RN}(\mathrm{MLP}(\mathbf{M}^{(t)})), \mathrm{RN}(\mathrm{MLP}(\mathbf{X}^{(t)}, \mathbf{Y}^{(t)})), \tag{4}$$

where $\mathrm{MLP}()$ and $\mathrm{RN}()$ are corresponding MLP layer and ResNet block, respectively. Then, an element-wise summation is used to fuse the two embeddings: $\mathbf{F}_{in}^{(t)} = \mathbf{F}_g^{(t)} + \mathbf{F}_f^{(t)}$. This operation can provide rich information for better learning of CL module from different aspects.

In turn, CL module offers effective keypoint feedback to KL module. On the one hand, we utilize the learned inlier weights to further enhance the representation ability of keypoints. Inspired by (Sarlin et al. 2020), KL module adopts a novel dynamic keypoint encoder to encode keypoint information into high dimensional features. Besides the information provided by feature detectors, we integrate the embeddings learned from CL module to enrich the descriptors:

$$\hat{\mathbf{x}}_i^{(t-1)} = \mathbf{x}_i^{(t-1)} + \mathrm{MLP}_{enc}(\mathbf{k}_i \,||\, \mathbf{p}_i^{(t-1)}). \tag{5}$$

where $||$ denotes the concatenation operation. $\mathrm{MLP}_{enc}(\cdot)$ is used to transform the feature dimension. When $t = 1$, we only utilize initial keypoint information $\mathbf{k}_i$, where $\mathbf{x}_i^{(0)} = \mathbf{d}_i^{(0)}$. And $\mathbf{p}_i^{(t-1)}$ is the inlier probability in the $(t-1)$-th iteration to represent the confidence of keypoint as inlier. Injecting this learned adjustment makes keypoints with similar confidence have similar input information, helping to better distinguish these features. The element-wise summation operation combines the features for descriptor augmentation. The augmented position-embedded feature $\hat{\mathbf{X}}^{(t-1)}$ and

$\hat{\mathbf{Y}}^{(t-1)}$ enable the following graph network to reason about both appearance and position jointly via the attention mechanism. On the other hand, some uninformative keypoints, which are usually not in the overlapping region of matching images, may cause the negative influence for network learning (Liu et al. 2023). Therefore, we propose an intuitive and effective adaptive sampling strategy to avoid updating these keypoints. This strategy can generate a mask score for each keypoint based on the inlier probability:

$$\mathbf{Ms}^{(t)}(\mathbf{x}_i) = \begin{cases} 1, & \mathbf{P}^{(t)}(\mathbf{x}_i) > \epsilon_m; \\ 0, & \mathbf{P}^{(t)}(\mathbf{x}_i) < \epsilon_m, \end{cases} \tag{6}$$

where ϵ_m is a sampling threshold to mitigate the over-pruning problem. Keypoints with low inlier weights will not participate in subsequent network learning, avoiding the computation and interference of uninformative keypoints.

3.3 Keypoint Learning Module

When matching a given ambiguous keypoint, people often repeatedly look back and forth between the two images. They need to search for and examine helpful contextual clues to recognize the correct matching keypoints (Chun 2000). This indicates that matching keypoints is an iterative process, requiring attention to be focused on specific locations (Sarlin et al. 2020). Here, KL module utilizes the self and cross attention with mask score to explore the context and enhance the representation of keypoints.

Specifically, given the input $\mathbf{X}^{(t-1)}$ and $\mathbf{Y}^{(t-1)}$, we first construct the complete intra-graph (within images) and inter-graph (between images) as (using $\mathbf{X}$ as an example):

$$\mathbf{X}^{(t)} = \hat{\mathbf{X}}^{(t-1)} + \mathbf{F}^{(t)}_{\mathrm{XS}} + \mathbf{F}^{(t)}_{\mathrm{XC}}, \tag{7}$$

$$\mathbf{F}^{(t)}_{\mathrm{XS}} = \mathrm{MLP}_S(FC(\mathbf{Att}^{(t)}_{\mathrm{XS}}) \,||\, \hat{\mathbf{X}}^{(t-1)}), \tag{8}$$

$$\mathbf{F}^{(t)}_{\mathrm{XC}} = \mathrm{MLP}_C(FC(\mathbf{Att}^{(t)}_{\mathrm{XC}}) \,||\, \hat{\mathbf{X}}^{(t-1)}). \tag{9}$$

$\mathbf{X}^{(t)}$ is the renewed descriptor of keypoints in $\mathrm{I_x}$ at the t-th iteration. $\mathbf{F}^{(t)}_{\mathrm{XS}}$ and $\mathbf{F}^{(t)}_{\mathrm{XC}}$ are self and cross attention features. $\mathrm{MLP}_{S/C}(\cdot)$ are 3-layer MLPs. $FC(\cdot)$ denotes a fully connected layer. $\mathbf{Att}^{(t)}_{\mathrm{XS}}$ and $\mathbf{Att}^{(t)}_{\mathrm{XC}}$ are self and cross attention contexts using mask score of each keypoints:

$$\mathbf{Q}^{(t)}_{\mathrm{X}}, \mathbf{K}^{(t)}_{\mathrm{X}}, \mathbf{V}^{(t)}_{\mathrm{X}} = FC(\hat{\mathbf{X}}^{(t-1)}), \tag{10}$$

$$\mathbf{Att}^{(t)}_{\mathrm{XS}} = sm(\frac{\mathbf{Ms}^{(t)}(\mathbf{X}^{(t-1)})\mathbf{Q}^{(t)}_{\mathrm{X}}(\mathbf{K}^{(t)}_{\mathrm{X}})^T}{\sqrt{d}})\mathbf{V}^{(t)}_{\mathrm{X}}, \tag{11}$$

$$\mathbf{Att}^{(t)}_{\mathrm{XC}} = sm(\frac{\mathbf{Ms}^{(t)}(\mathbf{X}^{(t-1)})\mathbf{Q}^{(t)}_{\mathrm{X}}(\mathbf{K}^{(t)}_{\mathrm{Y}})^T}{\sqrt{d}})\mathbf{V}^{(t)}_{\mathrm{Y}}. \tag{12}$$

$sm(\cdot)$ represents the row-wise softmax function. Mask score $\mathbf{Ms}^{(t)}$ is used for avoiding the negative influence of uninformative keypoints. For simplicity, we omit the formulation of multi-head attention, which can enhance the representation of features. Meanwhile, we adopt the same operation to obtain local feature $\mathbf{Y}^{(t)}$. The intra-graph and inter-graph enable each keypoint to be associated with other keypoints, while self and cross attention operations can selectively aggregate their contexts. Therefore, the enhanced feature $\mathbf{X}^{(t)}$ and $\mathbf{Y}^{(t)}$ have more strong representation due to the mutual communication of keypoints.

3.4 Correspondence Learning Module

Previous feature matchers (Sarlin et al. 2020; Shi et al. 2022) directly utilize the Sinkhorn algorithm or NN search on $\mathbf{X}^{(t)}$ and $\mathbf{Y}^{(t)}$ to establish keypoint matches. However, the paired relationship between keypoints ($i.e.$, correspondence) is rarely explored further, which is also vital for distinguishing matches. For example, the inliers typically conform to consistent constraints, such as lengths and angles, while mismatches exhibit random distribution. In this work, leveraging the consistency, we present a CL module to further distinguish matches and provide reliable keypoint feedback.

CL module contains the input embeddings, several ResNet blocks (Yi et al. 2018), the local consensus block, and the learning blocks. ResNet block is a basic learning structure, containing two MLP layers, several normalization operations, and ReLU to process matches. To explore the rich local context of matches, neighbor consistency has been employed in previous match filter works (Liu et al. 2021; Liu and Yang 2023). Inspired by CLNet (Zhao et al. 2021), we design a new local consensus block via additional spatial similarity to seek reliable neighbors. Specifically, we first conduct a neighbor search space:

$$\mathbf{F}^{(t)}_{ns} = \mathbf{F}^{(t)} \odot \mathbf{S}. \tag{13}$$

$\mathbf{F}^{(t)} = \left\{\mathbf{f}^{(t)}_1, ..., \mathbf{f}^{(t)}_n\right\} \in \mathbb{R}^{n \times d_m}$ is the middle feature map to measure high-dimensional feature similarity. $\odot$ is the Hadamard product. Meanwhile, inherent low-dimensional spatial similarity $\mathbf{S}$ as a regulator is calculated as:

$$s_{ij} = \max(0, 1 - \frac{d^2_{ij}}{\epsilon^2_d}), \tag{14}$$

where $max(0, \cdot)$ is a non-negative operation. d_{ij} denotes the length difference between two matches based on the coordinates. Two matches with the d_{ij} smaller than the distance hyper-parameter ϵ_d are regarded spatially compatible. Local neighbors are determined by Euclidean distance on $\mathbf{F}^{(t)}_{ns}$. Then, we construct a local neighbor graphs $\mathcal{G}^{(t)}_i = \left\{\mathcal{V}^{(t)}_i, \mathcal{E}^{(t)}_i\right\}$ for each match $\mathbf{m}^{(t)}_i$. Nodes $\mathcal{V}^{(t)}_i = \left\{\mathbf{m}^{(t)}_{i1}, ..., \mathbf{m}^{(t)}_{ik}\right\}$ represent the k-nearest neighbors of $\mathbf{m}^{(t)}_i$. Directed edges $\mathcal{E}_i = \left\{\mathbf{e}^{(t)}_{i1}, ..., \mathbf{e}^{(t)}_{ik}\right\}$ link $\mathbf{m}^{(t)}_i$ and its neighbors in $\mathcal{V}_i$ based on $\mathbf{F}^{(t)}$. The edge is built as:

$$\mathbf{e}^{(t)}_{ij} = (\mathbf{f}^{(t)}_i \,||\, \mathbf{f}^{(t)}_i - \mathbf{f}^{(t)}_{ij}), j = 1, 2, ..., k. \tag{15}$$

$\mathbf{f}^{(t)}_i, \mathbf{f}^{(t)}_{ij}$ are feature maps of $\mathbf{m}_i$ and its j-th neighbor $\mathbf{m}_{ij}$. Thus, we can obtain the neighbor embedding $\mathcal{G}^{(t)} \in \mathbf{R}^{n \times k \times 2d_m}$ of all matches. Then, local context can be aggregated in a grouped manner (Zhao et al. 2021):

$$\mathbf{F}^{(t)}_l = (Conv_2(Conv_1(\mathcal{G}^{(t)})), \tag{16}$$

Dataset	YFCC100M						Scannet					
Feature	RootSIFT			SuperPoint			RootSIFT			SuperPoint		
Matcher	@5°	@10°	@20°	@5°	@10°	@20°	@5°	@10°	@20°	@5°	@10°	@20°
NN-RT/MNN	26.7	43.2	59.4	6.5	15.4	28.5	9.1	19.8	32.7	9.4	21.6	36.4
AdaLAM	27.5	44.5	60.5	20.8	36.5	51.9	8.2	18.6	31.0	6.7	15.8	27.4
OANet	22.4	36.3	50.3	19.2	34.5	50.3	10.7	23.1	37.4	10.0	25.1	38.0
NCMNet	34.3	53.5	70.2	27.7	46.1	63.5	9.5	21.7	35.8	6.0	14.0	25.7
DeMatch	33.1	52.2	68.6	21.4	40.2	59.7	-	-	-	-	-	-
SuperGlue*	-	-	-	39.9	60.5	76.4	-	-	-	16.2	32.6	49.3
SuperGlue	35.1	54.2	70.9	33.2	53.5	70.8	14.7	29.4	45.6	12.0	26.3	42.4
SGMNet	34.8	54.1	70.9	33.0	53.0	70.0	14.4	29.9	46.0	16.4	32.1	48.7
LightGlue	35.8	55.3	72.0	39.5	59.5	75.5	15.5	31.0	47.0	15.4	31.2	47.5
EIMP	36.8	56.3	72.8	37.9	57.9	74.0	15.3	30.8	46.6	15.9	32.4	48.9
IMP	36.7	56.6	72.9	39.4	59.4	75.2	15.6	30.9	47.4	16.6	33.1	49.4
ECFM	**37.4**	**57.3**	**73.5**	41.9	61.7	77.0	16.2	32.2	48.6	17.2	33.6	49.4
CFM	37.1	56.9	73.2	**43.0**	**62.6**	**77.7**	**16.5**	**32.7**	**49.2**	**17.8**	**34.6**	**50.7**

Table 1: Quantitative results on outdoor YFCC100M and indoor Scannet dataset. **Bold** indicates the best.

where $Conv_1(\cdot)$ and $Conv_2(\cdot)$ denote the successive convolution layers with $1 \times \frac{k}{g}$ kernels and $1 \times g$ kernels, respectively. g is the number of groups. $\mathbf{F}_l^{(t)} \in \mathbb{R}^{n\times 1\times d_m}$ represents the local context feature. Then, the learning blocks contain several ResNet blocks and a global consensus block (Zhao et al. 2021) to encode global contextual information for getting the output feature map $\mathbf{F}_{out}^{(t)}$. Finally, the feature is processed to obtain the inlier probability set $\mathbf{P}$:

$$\mathbf{P}^{(t)} = \tanh(\mathrm{ReLU}(\mathrm{MLP}(\mathbf{F}_{out}^{(t)}))) \in [0, 1), \tag{17}$$

in which $\mathbf{F}_{out}^{(t)}$ is the output feature of the CL module. $\tanh(\cdot)$ and $\mathrm{ReLU}(\cdot)$ represent activation functions. $\mathbf{P}^{(t)} \in \mathbb{R}^{n\times 1}$ denotes the inlier weight, which indicates the probability of each initial match as an inlier.

3.5 Loss Function

We utilize keypoint learning loss $\mathcal{L}_k$ and correspondence learning loss $\mathcal{L}_m$ to optimize each KL module and CL module, respectively. $\mathcal{L}_k$ guarantees that keypoints possessing more discriminative descriptors attain higher matching scores. As (Sarlin et al. 2020), we use classification loss to minimize the negative log-likelihood of matching matrix:

$$\mathcal{L}_k = -\sum_{(i,j)\in\mathcal{P}^{gt}} \log \bar{\mathcal{P}}_{i,j} - \sum_{i\in\bar{\mathcal{P}}^{gt}} \log \bar{\mathcal{P}}_{i,n+1} - \sum_{j\in\bar{\mathcal{P}}^{gt}} \log \bar{\mathcal{P}}_{m+1,j}. \tag{18}$$

$\mathcal{P}^{gt}$ is the ground-truth matching matrix, and $\bar{\mathcal{P}}^{gt}$ is its expansion including an additional row and column. $\bar{\mathcal{P}} \in \mathbb{R}^{m+1\times n+1}$ is the matching matrix computed by the distance of descriptors. $\mathcal{L}_m$ aims to accurately distinguish correct matches, which is a binary cross entropy loss:

$$\mathcal{L}_m = H(\tau \odot \mathbf{P}, \mathbf{L}), \tag{19}$$

where $\mathbf{L}$ is the ground-truth label determined by epipolar distances. τ is temperature vector to alleviate label ambiguity. The final loss formulates as: $\mathcal{L} = \mathcal{L}_k + \mathcal{L}_m$. We apply $\mathcal{L}$ to each iteration and compute the average loss.

4 Experiments

4.1 Evaluation Protocols

Datasets. We construct experiments on relative pose estimation task to evaluate the performance on different matching scenes. The outdoor scene is Yahoo's YFCC100M (Thomee et al. 2016) dataset containing challenging tourist images. The indoor scene is the ScanNet (Dai et al. 2017) dataset consisting of large-scale RGB-D image pairs. Furthermore, we use the Aachen Day-Night (v1.0 and v1.1) (Sattler et al. 2018) datasets for visual localization task.

Evaluation. To obtain the relative pose, we estimate essential matrix to recover the rotation and translation vectors. The angular differences between ground truth vectors and estimated ones are selected as error metrics. Following benchmark (Sarlin et al. 2020), the cumulative error curve (AUC) at different thresholds (5°, 10°, and 20°) is the core default metric. For visual localization task, we use the percentage of correctly localized queries under the thresholds (0.25m/2°, 0.5m/5°, and 5m/10°) as evaluation metric.

4.2 Implementation Details

Following IMP, we train the network models on the MegaDepth (Li and Snavely 2018) dataset from scratch. It contains large-scale tourism landmarks collected from the Internet. Adam (Kingma and Ba 2014) optimizer with 500k iterations and a batchsize of 16 is used for optimizing network models. The initial learning rate is set as 10^{-4}. After 200k iterations, the learning rate decreases by a factor of 0.999992 and then remains fixed at 10^{-6}. Traditional RootSIFT and learning-based SuperPoint have been selected to extract 1024 feature keypoints as network inputs. Our KL module is executed T(9) times, while CL module runs only during the 6th and 9th iterations to avoid excessive overhead. The head in attention of KL module is set to 4. The feature dimension d_m, neighbor number k, group number g, distance hyper-parameter ϵ_d, and sampling threshold ϵ_m in CL module are experientially set as 128, 9, 3, 0.2, and -0.9.

Aachen v1.0	Day	Night
MNN	85.4 / 93.3 / 97.2	75.5 / 86.7 / 92.9
SuperGlue*	**89.6** / 95.4 / **98.8**	86.7 / 93.9 / **100.0**
SuperGlue	87.3 / 94.4/ 97.3	84.7 / 92.9 / 99.0
SGMNet	86.8 / 94.2 / 97.7	83.7 / 91.8 / 99.0
IMP	87.5 / 94.4 / 98.3	**87.8** / 93.9 / **100.0**
ECFM	88.5 / **95.5** / 98.2	**87.8** / **94.9** / **100.0**
CFM	89.4 / **95.5** / 98.4	**87.8** / 93.9 / **100.0**
Aachen v1.1	Day	Night
MNN	87.9 / 93.6 / 96.8	70.2 / 84.8 / 93.7
SuperGlue*	89.8 / **96.6** / **99.4**	**75.9** / 90.1 / **100.0**
SuperGlue	88.8 / 95.4 / 98.7	75.0 / 91.1 / 97.4
SGMNet	88.7 / 96.2 / 98.9	**75.9** / 89.0 / 99.0
IMP	89.6 / 95.8 / 99.0	75.4 / 91.1 / 99.5
ECFM	**89.9** / 95.8 / 99.0	**75.9** / **91.6** / 99.5
CFM	89.6 / 96.0 / 99.0	75.4 / **91.6** / 99.5

Table 2: Results with different thresholds (0.25m/2°, 0.5m/5°, and 5m/10°) on Aachen v1.0 and v1.1 dataset.

4.3 Comparative Results

Baselines: We compare CFM with state-of-the-art detector-based feature matchers, including SuperGlue (Sarlin et al. 2020), SGMNet (Chen et al. 2021), LightGlue (Lindenberger, Sarlin, and Pollefeys 2023), and IMP (Xue, Budvytis, and Cipolla 2023), and match filters, $i.e.$, AdaLAM (Cavalli et al. 2020), OANet (Zhang et al. 2019), NCMNet (Liu and Yang 2023), and DeMatch (Zhang et al. 2024), as well as several existing detector-free methods (LoFTR (Sun et al. 2021), PDC-Net+ (Truong et al. 2023), and DKM (Edstedt et al. 2023)). We show the results of original model (SuperGlue*) and the retrained model by SGMNet (SuperGlue). NN search via descriptors with ratio test or mutual check is used to obtain feature matches. The match filters take matches estimated by the NN search as inputs.

Relative Pose Estimation. Table 1 shows the comparison results on YFCC100M. We can see that NN search and match filters generally perform worse than feature matchers, especially when SuperPoint is used to extract feature points. This is due to the limited generalization ability of the learning-based feature detector, so the quality of estimated initial matches solely using NN search is difficult to guarantee. Feature matchers can further improve the discrimination of keypoints by integrating both geometric information and descriptors. Therefore, when feature matchers are equipped, the performance improves dramatically, $e.g.$, AUC@5° from 6.5% of MNN to 33.2 % of SuperGlue. CFM and ECFM can obtain the best results in all settings by combining the advantage of keypoint and correspondence learning, achieving more precise relative poses. Furthermore, ECFM may remove some essential keypoints, resulting in a slight degradation when using SuperPoint, which is consistent with EIMP. However, our ECFM decreases significantly less than EIMP, proving the advantage of our adaptive sampling strategy.

The results on Scannet are reported in Table 1. Indoor scenes are more challenging than outdoor scenes due to repetitive structures and texture-less regions. Our methods show superior performance compared to the state-of-the-arts in both settings. Overall, these results highlight the generalization ability of our models to establish precise matches for the pose estimation. Furthermore, Fig. 3 exhibits some qualitative results on both challenging datasets. The proposed methods can consistently establish reliable matches when image pairs face occlusions, large viewpoint variations, repetitive structures, textureless objects, etc.

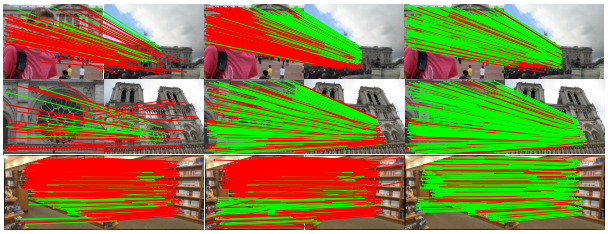


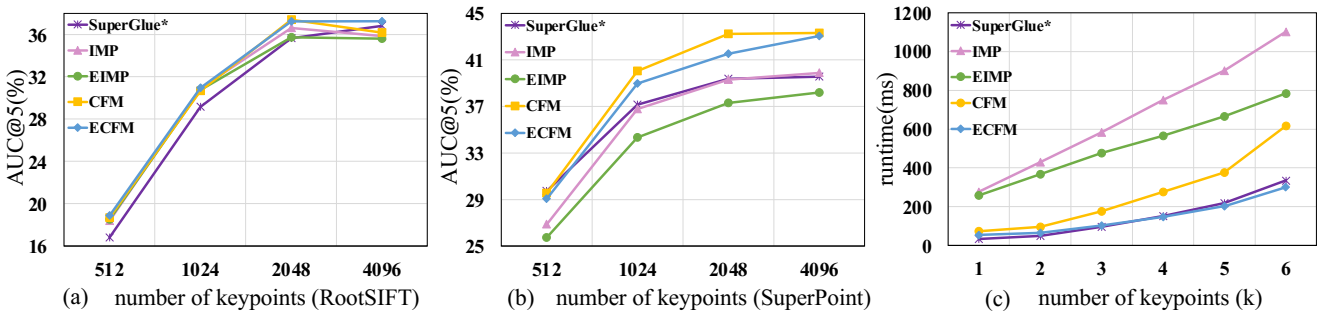
Figure 3: Visualization on the outdoor and indoor scenes. Correct matches (green) and mismatches (red) are exhibited.

Datasets	MegaDepth1500			YFCC100M			rt(ms)
Method	@5°	@10°	@20°	@5°	@10°	@20°	
SuerGlue(18)	44.8	62.5	76.4	34.5	54.6	71.7	**81**
LoFTR	52.8	69.2	81.2	39.8	60.0	76.1	189
PDC-Net+	51.5	67.2	78.5	37.5	58.1	74.5	740
DKM	**60.4**	**74.9**	**85.1**	44.1	63.7	78.4	655
CFM	54.0	68.2	79.2	43.0	62.6	77.7	96
HCFM	55.4	69.9	80.5	**44.3**	**64.0**	**78.6**	211

Table 3: Results on MegaDepth1500 and YFCC100M (some results are from Dematch) compared with dense methods. AUC and runtime are reported.

Visual Localization. Table 2 reports the comparative results on the Aachen dataset for large-scale localization. This task evaluates the robustness of methods when facing severe illumination changes. We can see that our CFM and ECFM are able to achieve competitive results on both scenes, especially for nighttime images, compared to other competitors. Noteworthily, SuperGlue* undergoes additional pretraining in the large-scale image retrieval benchmark (Radenović et al. 2018), obtaining significant gains than retained SuperGlue. Our models are solely trained on the MegaDepth dataset. This further proves the strong generalization ability of our methods on the real challenging application.

Comparisons with Dense Methods. To further verify the advantage of CFM, comparisons with more advanced detector-free methods are reported in Table 3. HCFM is a huge version of CFM by adding CL module after each KL module. Without relying on expensive pixel-wise learning, CFM and HCFM can achieve competitive (MegaDepth1500) or even better (YFCC100M) results compared with dense competitors. Moreover, our methods significantly outperform SuperGlue(18) that contains 18 atten-



(a) number of keypoints (RootSIFT) (b) number of keypoints (SuperPoint) (c) number of keypoints (k)

Figure 4: Performance comparisons under the different number of keypoints, where feature keypoints are extracted by (a) traditional RootSIFT and (b) learning-based SuperPoint. Furthermore, (c) runtime is also reported.

Methods	@5°	@10°	@20°
SuperGlue* + NCMNet	35.3	55.8	73.3
SuperGlue* + CLNet	35.2	55.8	73.2
IMP + NCMNet	33.1	53.4	70.9
IMP + CLNet	32.1	52.6	71.1
SuperGlue* + CLNet* (re)	34.6	54.8	72.0
SuperGlue* + CLNet (re)	38.8	59.0	75.5

Table 4: Comparison of different combinations between feature matchers and match filters. **(re)** means that match filter is retrained using input matches provided by SuperGlue*.

KL	CL	FE	SS	KF	@5°	@10°	@20°
✓	✓				39.2	59.3	75.3
✓	✓	✓			40.5	60.6	76.2
✓	✓	✓	✓		41.1	61.3	76.6
✓	✓	✓	✓	✓	43.0	62.5	77.7

Table 5: Ablation studies regarding the gains of main components. **FE**: the feature embedding provided by KL module. **SS**: the spatial similarity in local consensus block. **KF**: the keypoint feedback provided by CL module.

tion modules, highlighting the limitations of single keypoint learning. This also indicates the effectiveness and rationality of our collaborative learning.

4.4 Ablation Studies

We further construct ablation studies to examine the effects of each component in CFM on YFCC100M dataset.

Main components. Table 4 presents the results of combinations between off-the-shelf feature matcher and match filter. CLNet* additionally adds the feature embedding as CFM. We observe that the latter does not further improve the former, even after re-training the matching filters, which demonstrates the information incompatibility between the two methods. In our work, KL module is used for obtaining distinctive descriptors and providing high-quality input matches. And CL module is designed to capture the rich context of matches and give effective keypoint feedback. Here, we evaluate the performance gains of each main component in CFM as shown in Table 5. The gap between them can be alleviated by joint optimization, achieving better results (38.8% reported in Table 4 vs 43.0% AUC@5°). At the same time, the embeddings learned from both modules can be used with each other in a progressive manner. Therefore, when CL module is equipped with the FE and SS, and KL module uses the KF, the performance will be further improved owing to the information interaction. Notably, with the aid of KF, the gains are more significant. This further demonstrates the feasibility and effectiveness of using correspondence learning to provide reliable feedback.

The keypoint number. This is critical for feature matching, affecting both accuracy and speed. Here, we present comparisons with several methods for the different number of RootSIFT and SuperPoint keypoints as shown in Fig. 4 (a) and (b), respectively. As expected, the performance of all methods improves with the increasing number of keypoints. Our methods show notable performance gains over other competitors, thanks to the collaborative effect. Notably, when using RootSIFT, ECFM outperforms CFM, primarily due to the removal of uninformative keypoints. Meanwhile, ECFM obtains more significant gains as the keypoints increase. Moreover, Fig. 4 (c) presents the runtime comparisons. IMP and EIMP are significantly slower than other methods since they require camera pose computation at each iteration. Our ECFM shows significant efficiency benefiting from the adaptive sampling strategy compared to CFM. Especially, when using more keypoints, ECFM works faster than SuperGlue*, indicating its potential in efficiency. Overall, our methods achieve the best performance and competitive efficiency across different number of keypoints.

5 Conclusion

In this paper, we develop a collaborative feature matching (CFM), including KL module and CL module, to bridge the gap of previous two methods. These two modules enable mutual enhancement in a progressive manner. With the collaborative effect, our method can acquire more reliable feature matches and recover accurate relative poses under challenging matching scenes. Experimental results on different benchmarks indicate that the proposed methods outperform state-of-the-art feature matching competitors.

Acknowledgments

This work was supported by the Natural Science Foundation of Tianjin, China (No.24JCZXJC00040), Shenzhen Science and Technology Program (No. JCYJ20240813114229039), the National Natural Science Foundation of China (No. 623B2056, 624B2072), the Fundamental Research Funds for the Central Universities, the Supercomputing Center of Nankai University (NKSC).

References

Cavalli, L.; Larsson, V.; Oswald, M. R.; Sattler, T.; and Pollefeys, M. 2020. Handcrafted outlier detection revisited. In *Proceedings of the European Conference on Computer Vision*, 770–787.

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

Chun, M. M. 2000. Contextual cueing of visual attention. *Trends in cognitive sciences*, 4(5): 170–178.

Cuturi, M. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems*, 26.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5828–5839.

Dai, L.; Du, X.; and Tang, J. 2024. TrGa: Reconsidering the Application of Graph Neural Networks in Two-View Correspondence Pruning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 5633–5642.

Dai, L.; Du, X.; Zhang, H.; and Tang, J. 2024. Mgnet: Learning correspondences via multiple graphs. In *Proceedings of the AAAI conference on Artificial Intelligence*, 3945–3953.

Dai, L.; Liu, Y.; Ma, J.; Wei, L.; Lai, T.; Yang, C.; and Chen, R. 2022. MS2DG-Net: Progressive correspondence learning via multiple sparse semantics dynamic graph. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8973–8982.

DeTone, D.; Malisiewicz, T.; and Rabinovich, A. 2018. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 224–236.

Edstedt, J.; Athanasiadis, I.; Wadenbäck, M.; and Felsberg, M. 2023. DKM: Dense Kernelized Feature Matching for Geometry Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17765–17775.

Edstedt, J.; Sun, Q.; Bökman, G.; Wadenbäck, M.; and Felsberg, M. 2024. RoMa: Robust dense feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19790–19800.

He, J.; Gao, Y.; Zhang, T.; Zhang, Z.; and Wu, F. 2023. D2Former: Jointly Learning Hierarchical Detectors and Contextual Descriptors via Agent-Based Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2904–2914.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Jiang, H.; Karpur, A.; Cao, B.; Huang, Q.; and Araujo, A. 2024. OmniGlue: Generalizable Feature Matching with Foundation Model Guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19865–19875.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lee, J.; Kim, B.; Kim, S.; and Cho, M. 2023. Learning Rotation-Equivariant Features for Visual Correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21887–21897.

Li, Z.; and Snavely, N. 2018. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2041–2050.

Lindenberger, P.; Sarlin, P.-E.; and Pollefeys, M. 2023. LightGlue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 17627–17638.

Liu, X.; Qin, R.; Yan, J.; and Yang, J. 2024a. NCMNet: Neighbor Consistency Mining Network for Two-View Correspondence Pruning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12): 11254 – 11272.

Liu, X.; Xiao, G.; Chen, R.; and Ma, J. 2023. PGFNet: Preference-Guided Filtering Network for Two-View Correspondence Learning. *IEEE Transactions on Image Processing*, 32: 1367 – 1378.

Liu, X.; and Yang, J. 2023. Progressive Neighbor Consistency Mining for Correspondence Pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9527–9537.

Liu, Y.; Li, Y.; and Zhao, S. 2025. TransMatch: Transformer-based correspondence pruning via local and global consensus. *Pattern Recognition*, 159: 111120.

Liu, Y.; Liu, L.; Lin, C.; Dong, Z.; and Wang, W. 2021. Learnable motion coherence for correspondence pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3237–3246.

Liu, Y.; Zhao, B. N.; Zhao, S.; and Zhang, L. 2022. Progressive Motion Coherence for Remote Sensing Image Matching. *IEEE Transactions on Geoscience and Remote Sensing*, 60: 1–13.

Liu, Y.; Zhou, W.; Li, Y.; and Zhao, S. 2024b. RoSe: Rotation-invariant sequence-aware consensus for robust correspondence pruning. In *Proceedings of the ACM International Conference on Multimedia*, 7754–7763.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2): 91–110.

Ma, J.; Jiang, X.; Fan, A.; Jiang, J.; and Yan, J. 2021. Image matching from handcrafted to deep features: A survey. *International Journal of Computer Vision*, 129(1): 23–79.

Pautrat, R.; Suárez, I.; Yu, Y.; Pollefeys, M.; and Larsson, V. 2023. Gluestick: Robust image matching by sticking points and lines together. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9706–9716.

Potje, G.; Cadar, F.; Araujo, A.; Martins, R.; and Nascimento, E. R. 2023. Enhancing Deformable Local Features by Jointly Learning to Detect and Describe Keypoints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1306–1315.

Radenović, F.; Iscen, A.; Tolias, G.; Avrithis, Y.; and Chum, O. 2018. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5706–5715.

Ranftl, R.; and Koltun, V. 2018. Deep fundamental matrix estimation. In *Proceedings of the European Conference on Computer Vision*, 284–299.

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

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

Sattler, T.; Maddern, W.; Toft, C.; Torii, A.; Hammarstrand, L.; Stenborg, E.; Safari, D.; Okutomi, M.; Pollefeys, M.; Sivic, J.; et al. 2018. Benchmarking 6dof outdoor visual localization in changing conditions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8601–8610.

Shi, Y.; Cai, J.-X.; Shavit, Y.; Mu, T.-J.; Feng, W.; and Zhang, K. 2022. 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.

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

Thomee, B.; Shamma, D. A.; Friedland, G.; Elizalde, B.; Ni, K.; Poland, D.; Borth, D.; and Li, L.-J. 2016. YFCC100M: The new data in multimedia research. *Communications of the ACM*, 59(2): 64–73.

Truong, P.; Danelljan, M.; Timofte, R.; and Van Gool, L. 2023. Pdc-net+: Enhanced probabilistic dense correspondence network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8): 10247–10266.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008.

Wang, Y.; He, X.; Peng, S.; Tan, D.; and Zhou, X. 2024. Efficient LoFTR: Semi-dense local feature matching with sparse-like speed. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21666–21675.

Wu, L.; Cui, P.; Pei, J.; Zhao, L.; and Guo, X. 2022. Graph neural networks: foundation, frontiers and applications. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 4840–4841.

Xiao, G.; Liu, X.; Zhong, Z.; Zhang, X.; Ma, J.; and Ling, H. 2024. T-Net++: Effective Permutation-Equivariance Network for Two-View Correspondence Pruning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12): 10629 – 10644.

Xue, F.; Budvytis, I.; and Cipolla, R. 2023. IMP: Iterative Matching and Pose Estimation with Adaptive Pooling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21317–21326.

Ye, X.; Zhao, W.; Lu, H.; and Cao, Z. 2023. Learning Second-Order Attentive Context for Efficient Correspondence Pruning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 3250–3258.

Yi, K. M.; Trulls, E.; Lepetit, V.; and Fua, P. 2016. Lift: Learned invariant feature transform. In *Proceedings of the European Conference on Computer Vision*, 467–483.

Yi, K. M.; Trulls, E.; Ono, Y.; Lepetit, V.; Salzmann, M.; and Fua, P. 2018. Learning to find good correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2666–2674.

Zhang, J.; Sun, D.; Luo, Z.; Yao, A.; Zhou, L.; Shen, T.; Chen, Y.; Quan, L.; and Liao, H. 2019. Learning two-view correspondences and geometry using order-aware network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5845–5854.

Zhang, S.; Li, Z.; Gao, Y.; and Ma, J. 2024. DeMatch: Deep Decomposition of Motion Field for Two-View Correspondence Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20278–20287.

Zhao, C.; Ge, Y.; Zhu, F.; Zhao, R.; Li, H.; and Salzmann, M. 2021. Progressive correspondence pruning by consensus learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6464–6473.