NCMNet: Neighbor Consistency Mining Network for Two-View Correspondence Pruning

Xin Liu, Rong Qin, Junchi Yan, *Senior Member, IEEE*, and Jufeng Yang

Abstract—Correspondence pruning plays a crucial role in a variety of feature matching based tasks, which aims at identifying correct correspondences (inliers) from initial ones. Seeking consistent k-nearest neighbors in both coordinate and feature spaces is a prevalent strategy employed in previous approaches. However, the vicinity of an inlier contains numerous irregular false correspondences (outliers), which leads them to mistakenly become neighbors according to the similarity constraint of nearest neighbors. To tackle this issue, we propose a global-graph space to seek consistent neighbors with similar graph structures. This is achieved by using a global connected graph to explicitly render the affinity relationship between correspondences based on the spatial and feature consistency. Furthermore, to enhance the robustness of method for various matching scenes, we develop a neighbor consistency block to adequately leverage the potential of three types of neighbors. The consistency can be progressively mined by sequentially extracting intra-neighbor context and exploring inter-neighbor interactions. Ultimately, we present a Neighbor Consistency Mining Network (NCMNet) to estimate the parametric models and remove outliers. Extensive experimental results demonstrate that the proposed method outperforms other state-of-the-art methods on various benchmarks for two-view geometry estimation. Meanwhile, four extended tasks, including remote sensing image registration, point cloud registration, 3D reconstruction, and visual localization, are conducted to test the generalization ability. **The source code is provided in https://github.com/xinliu29/NCMNet.**

Index Terms—Correspondence pruning, feature matching, neighbor consistency, global-graph, parametric models

1 Introduction

Accurately estimating feature correspondences between image pairs is essential for various computer vision tasks, such as visual simultaneous localization and mapping [1], structure from motion [2], [3], image registration [4], [5], and visual localization [6]. Given a pair of images, feature keypoints and their corresponding descriptors can be obtained by employing existing feature extraction methods, including handcrafted [7], [8], [9] and learning-based works [10], [11], [12]. Then, we establish initial correspondences either by imposing a similarity constraint on descriptors or by utilizing advanced deep learning algorithms [7], [11], [13]. However, abundant false correspondences ($i.e.$, outliers) inevitably exist due to the limitation of local descriptors [14], [15], [16], especially when facing severe illumination changes, viewpoint variations, occlusions, blurs, etc. These outliers can significantly impact the accuracy of downstream feature matching based tasks. Hence, to alleviate this problem, correspondence pruning [17], [18], [19] is used to further recognize correct correspondences ($i.e.$, inliers) from initial ones.

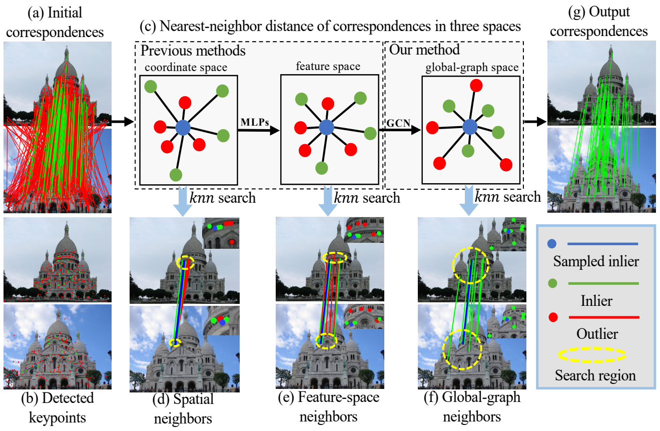

Fig. 1. The process of NCMNet. Initial correspondences (a) established by SIFT [7] feature keypoints (b) contain numerous outliers. As illustrated in (c), our global-graph space can decrease the nearest-neighbor distance of some inliers that are far from the sampled point in the other two spaces, making it possible for them to become neighbors. We exhibit three types of neighbors of a sampled inlier, including (d) spatial neighbors, (e) feature-space neighbors, and (f) global-graph neighbors. The neighbor search region is displayed with the yellow ellipse. Our NCMNet is able to obtain excellent results as shown in (g). **MLPs**: the Multi-Layer Perceptrons. **GCN**: the modified Graph Convolution Network.

As the pioneer, RANSAC [20] and its variants [21], [22], [23] employ a hypothesize-and-verify framework to seek an optimal parametric model with maximum supporters iteratively. Their stability gradually decreases as the inlier ratio decreases, primarily due to the adverse impact of numerous outliers on model generation. In addition to searching for possible models in a greedy manner, there are plenty of efforts leveraging the geometric property of correspondence [15], [17], [24]. As illustrated in Fig. 1 (a), the distribution of inliers and outliers has significant differences under the 2D rigid transformation. That is, inliers commonly conform to consistent constraints ($e.g.$, similar lengths, angles, and motion), while outliers are randomly distributed. Therefore, correspondence consistency is considered as vital priori knowledge, and has been extensively studied to distinguish inliers from outliers [14], [25].

- *X. Liu, R. Qin and J. Yang are with the VCIP & TMCC & DISSec, College of Computer Science, Nankai University, and Nankai International Advanced Research Institute (SHENZHEN·FUTIAN). J. Yang is also with Pengcheng Laboratory, Shenzhen, China. (E-mail: xinliu_0209@163.com, qinrong_nk@mail.nankai.edu.cn, yangjufeng@nankai.edu.cn).*
- *J. Yan is with the Department of Computer Science and Engineering, MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200240, China. (E-mail: yanjunchi@sjtu.edu.cn).*

Neighbor consistency has received significant attention due to its efficiency, which only focuses on few elements for each correspondence. For well-defined neighbors, earlier studies [17], [26], [16] utilize k-nearest neighbor (knn) search within the coordinate space of original correspondences to find spatially consistent correspondences, known as spatial neighbors. In recent years, some learning-based works [27], [28] seek feature-consistent correspondences, called feature-space neighbors, using knn search in the high-dimensional feature space derived from network learning. They demonstrate the promising advancements of neighbor consistency for distinguishing correspondences. Nonetheless, initial correspondences are often distributed unevenly across image pairs, which can lead to the presence of numerous random outliers in the vicinity of inliers, particularly in wide-baseline scenarios with approximately 90% outliers [29], [27]. As a result, some correspondences mistakenly become neighbors since they are close to each other in the above two Euclidean spaces as illustrated in Fig. 1 (c). As shown in Fig. 1 (d-e), in the coordinate and feature spaces, the searched neighbors of a sampled inlier (blue line) contain some unexpected outliers (red line). In fact, it's quite challenging to handle this situation only through similarity constraints in Euclidean space [15], [30].

To tackle the above problem, we present a non-Euclidean global-graph space. In particular, inliers tend to be consistent at the global level [20], [15], [30]. That is, a sampled inlier has strong connections with other inliers, and weak or no connections with outliers. They are able to form similar graph structures [31], [32], [33], [34], which can be well recognized by Graph Convolution Network (GCN) [35]. Therefore, we capture this global consistency by constructing a graph space where the neighbor definition depends on the similarity of graph structure. We then adopt a modified GCN to further explore this consistency and enhance the long-range dependencies among correspondences. Compared to the previous Euclidean spaces, the inliers have strong affinity relationship in global-graph space. Thus, our global-graph space is able to pull the distance of correspondences with similar graph structures. These correspondences may be difficult to become neighbors since their nearest-neighbor distance is large in the coordinate and feature spaces as shown in Fig. 1 (c). More specifically, we start by constructing a weighted global graph, where nodes represent all correspondences, and edges denote their pairwise affinities calculated using the consistency scores. To obtain a more representative graph, we develop a spatial consistency based on the length constraint for complementing the feature consistency used in [36]. Next, we utilize a modified GCN [35] to obtain our global-graph space. Ultimately, we employ knn search within this space to identify globally consistent correspondences, referred to as global-graph neighbors. Noteworthily, the global-graph neighbors are not spatially close to the sampled correspondence as illustrated in Fig. 1(f). In other words, it has a larger search region (refer to the ablation) owing to our global-graph operation.

The spatial and feature-space neighbors, which concentrate on the local scope of sampled correspondence, are searched by the low-dimensional spatial similarity and high-dimensional feature similarity, respectively. In contrast, our global-graph neighbors focus on globally consistent neighbors with similar graph structures. As described in [37], [27], correspondence pruning requires rich local and global contexts. Therefore, to enhance the robustness for challenging matching scenarios, we design a neighbor consistency (NC) block to fully leverage the potential of three types of neighbors. NC block contains three essential components: neighbor embedding construction, self-context extraction (SCE) layer, and cross-context interaction (CCI) layer. To be specific, we first construct three directed graphs according to different neighbors as neighbor embeddings To extract corresponding neighbor context features, SCE layer dynamically captures intra-neighbor relationships and aggregates their contextual information using a grouped convolution manner. CCI layer serves to further explore their interactions. Due to the limited capacity of single cross-attention branch used in [36], we design a hierarchical grouped manner to effectively fuse and modulate inter-neighbor interactive information. Building on NC block, we propose Neighbor Consistency Mining Network (NCMNet) [36] and NCMNet+ with two improvements to achieve two-view correspondence pruning.

The contributions are three-fold: (1) We propose a global-graph space by constructing explicit connections among correspondences via the spatial and feature consistency. It is used for seeking consistent neighbors with similar graph structures. (2) Leveraging the potential of three types of neighbors, we develop the NC block to progressively mine the neighbor consistency through the extraction of intra-neighbor context as well as the exploration of inter-neighbor interactions. (3) We prove the effectiveness and generalization ability of our approach by a series of geometry estimation benchmarks and extended tasks.

This manuscript is an extension of our preliminary conference version [36] appeared in CVPR 2023 with the following improvements: (1) In the construction of weighted global graph, NCMNet relies solely on feature consistency learned from networks, which may be inaccurate due to the ambiguity of learning process. NCMNet+ applies the spatial consistency inherent in correspondences to complement the feature consistency and enhance the reliability of global-graph space. (2) In the CCI layer, NCMNet uses a single cross-attention operation to explore inter-neighbor interaction. On this basis, NCMNet+ utilizes an effective hierarchical grouped manner to enrich information integration, thereby improving correspondence pruning accuracy. (3) In terms of validation, NCMNet focuses on estimating the essential matrix to recover camera poses. We add the estimation of both fundamental matrix and homography matrix in this work. Meanwhile, we conduct extensive experiments and thorough ablation analysis to fully comprehend our method. (4) We further extend the proposed method to tackle four feature matching based tasks, including remote sensing image registration, point cloud registration, 3D reconstruction, and visual localization.

2 Related Work

Generally, feature matching works can be classified into two major directions: detector-free and detector-based approaches. Detector-free methods [38], [39], [40], [41], [42] directly process the image pairs and generate pixel-wise dense correspondences. While these techniques are powerful, they often come with a significant computational overhead due to the extremely large number of pixels in images. In contrast, detector-based methods [7], [10], [43], [44] have played a key role in the past decades, which construct point-wise sparse correspondences by detecting distinctive keypoints, and then match them. However, looking for accurate feature correspondences is still complicated due to the unbalanced distribution between inliers and outliers [45], [46]. This problem can be mitigated by further applying correspondence pruning methods. In the subsequent sections, we will provide a detailed review for the relevant background materials in detail.

2.1 RANSAC-Related Methods

As one of the the most well-known algorithms in recent decades, RANSAC [20] applies a hypothesize-and-verify framework to find the largest inlier set. More specifically, it randomly selects a minimal subset of data to generate a hypothetical parametric model, $e.g.$, 5 correspondences for essential matrix [47]. Then, the reliability of model can be verified by the number of correspondences that conform to the model. This process will continue until it reaches the predefined iterations or thresholds. Following this framework, subsequent works [21], [22], [23], [48] improve either efficiency or effectiveness using different sampling and verification strategies. MLESAC [21] determines the optimal model through maximizing the log likelihood of correspondences to enhance the algorithm's robustness. USAC [23] reviews the related variants and presents a universal structure based on some important considerations, showing better improvements. MAGSAC [48] achieves superior performance by utilizing σ-consensus to avoid the predefined inlier-outlier threshold. Furthermore, some variations [49], [50], [51], [52], [53], [54] leverage deep learning pipelines to perfect the quality of parametric models. These RANSAC-related works continue to be regarded as standard solutions for outlier removal and parametric model estimation. However, this random sampling strategy is sensitive to outliers [45], [55], [56]. Their performance drops significantly as the proportion of outliers in initial correspondences gradually increases.

2.2 Learning-Based Methods

The development of deep learning technologies has led to several pioneer works [49], [57], [18] that utilize neural networks to accomplish correspondence pruning. For example, DSAC [49] devises a differentiable counterpart of RANSAC based on the probabilistic selection. Recently, PointNets [58], [59] utilize Multi-Layer Perceptrons (MLPs) to deal with unordered and irregular point sets, and have gained widespread attention. Drawing inspiration from PointNets, LFGC [18] trains a permutation-equivariant structure via MLPs to estimate inlier weights of correspondences and regress camera poses encoded by the essential matrix. Similarly, DFE [57] also adopts deep networks to predict inlier weights that are used for fundamental matrix estimation.

Follow-up methods take this correspondence classification paradigm as a de facto standard and improve the performance in different manners. On the one hand, several works devise diverse network structures to capture rich contextual information. To obtain the local context, OANet [37] learns the soft assignment matrix by a differentiable pooling layer for clustering input correspondences. Then, it recovers the original size of correspondences using an upsampling operation. It also transposes the dimension of features to exploit the global context. T-Net [60] introduces a T-shaped network architecture to integrate the outputs of iterative sub-networks. ConvMatch [61] develops a regular motion field and explores the possibility of using the convolutional neural network to capture context. On the other hand, some researchers take advantage of attention mechanism [62] to improve the representation of crucial features. ACNe [63] leverages attention weights for normalizing the features from local and global aspects. ANA-Net [64] calculates the similarity of attention weights for discovering attention-consistent correspondences. The multi-scale attention offered by MSA-Net [65] and the grouped residual attention used in PGFNet [66] are able to further improve the accuracy of correspondence pruning. Although the above-mentioned works have exhibited outstanding performance, they still suffer from some limitations. First, it is not intuitive to implicitly capture context by designing data-independent operators on correspondence learning. Second, outliers will severely hinder network learning and convergence. Most of them are still susceptible to the negative impact of high ratio outliers during the network training though attention mechanism aims to alleviate this problem. Unlike these methods, we adequately leverage different types of neighbor consistency to explicitly incorporate the intrinsic geometric and feature property of correspondences into the network learning process. We also utilize the iterative pruning strategy [27] as the basic framework to extract more reliable candidates for better network learning.

2.3 Consistency of Correspondences

Under the 2D rigid transformation, inliers usually have consistent constraints while the distribution of outliers is random [14]. Therefore, consistency of correspondences is an important clue to separate inliers and outliers, which has been studied extensively in the past [67], [45], [68]. For example, BF [15] formulates the piecewise consistency constraint via the proposed bilateral functions for global modeling. CODE [30] designs a consistent separability constraint at a global level to filter highly noisy correspondences. GMS [17] seeks consistent spatial neighbors to determine the reliability of correspondences by a grid-based score estimator. LPM [69] also explores neighbor consistency using predefined statistical measures. These handcrafted methods need elaborate parameter tuning to achieve satisfactory performance, meanwhile, they are sensitive to challenging matching scenes such as large viewpoint changes [45], [46].

Inspired by these traditional technologies, some works explore the consistency of correspondences in a learnable manner. NM-Net [29] develops a hierarchical network to mine the context of compatibility-specific neighbors based on local affine information [70]. LMCNet [16] reformulates motion coherence of spatial neighbors into a smooth function solved by graph Laplacian. CLNet [27] searches for neighbors in the feature space, and designs a local-to-global consensus learning framework. MS^{2}DG-Net [28] also exploits the local topology of feature-space neighbors by semantics dynamic graph. They aggregate neighbor information by various network structures or learning paradigms for robust correspondence pruning. However, as analyzed in Section 1, the neighbors searched from above-mentioned spaces may be inconsistent because of the irregular distribution of numerous outliers. In this paper, we propose a global-graph space to explicitly capture the strong consistency of inliers at a global level so that correspondences with similar global graph structures can be neighbors. Meanwhile, to enhance the robustness of method for complex matching scenes, we empirically devise a neighbor consistency block. It progressively extracts and integrates three types of neighbor contexts through the proposed SCE layer and CCI layer.

3 Methodology

This section, we will describe the details of our method. The problem formulation of two-view correspondence pruning is first described. The details of NCMNet, including global-graph space, neighbor consistency block, and network architecture, are then introduced. Finally, we give the description of loss function.

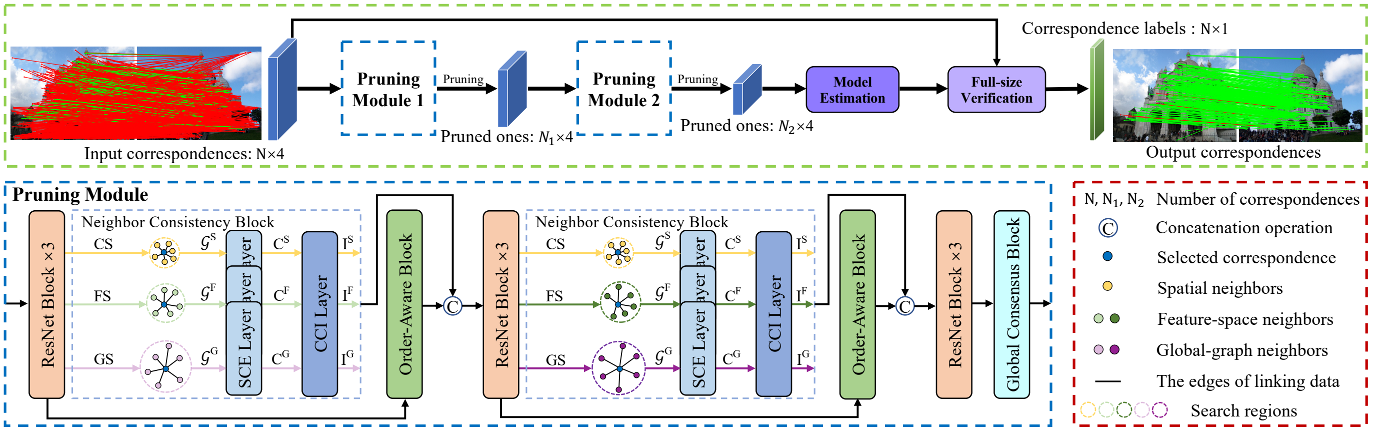

Fig. 2. Framework of our NCMNet. $N \times 4$ initial correspondences are established as inputs, then, the parametric model and $N \times 1$ inlier probabilities are estimated. The iterative pruning strategy containing two pruning modules is adopted as the core architecture to distill more reliable candidates for model estimation. Each pruning module includes several existing network structures and the proposed Neighbor Consistency (NC) block. NC block mainly consists of three key parts: the construction of three neighbor embeddings ($\mathcal{G}^S, \mathcal{G}^F, \mathcal{G}^G$), Self-Context Extraction (SCE) layer to capture and aggregate intra-neighbor context (C^S, C^F, C^G), and Cross-Context Interaction (CCI) layer to fuse and modulate inter-neighbor information (I^S, I^F, I^G). **CS**: the coordinate space, **FS**: the feature space, **GS**: the global-graph space.

3.1 Problem Formulation

Given a pair of matching images, we can utilize handcrafted feature extraction methods [7], [9] or learned ones [10], [11] to obtain feature keypoints and associated descriptors. The primary correspondence set $S = \{s_1, s_2, ..., s_N\} \in \mathbb{R}^{N \times 4}$ can be estimated by the similarity matching strategy of descriptors or neural networks [71]. Here, $s_i = (u_i, v_i)$ denotes the i-th correspondence, where u_i and v_i are normalized keypoint coordinates using camera intrinsics in two matching images, respectively. N denotes the number of initial correspondences. However, the outliers caused by the ambiguity of feature descriptors are inevitable. Thus, the purpose of our correspondence pruning is to filter out outliers from initial correspondences.

To achieve this purpose, the correspondence pruning process usually takes the initial correspondence set S as **input**, and **outputs** the label ($i.e.$, outlier or inlier) of all correspondences. That is, the set S is split into an inlier set S_{in} and an outlier set S_{out}. Meanwhile, the parametric models ($e.g.$, essential matrix) are estimated by using the S_{in} to evaluate the performance of methods. The parametric model is utilized to recover the camera poses of matching images, including corresponding rotation and translation vectors. In summary, the optimization objective of a correspondence pruning method is seeking sufficient inliers to recover accurate camera poses.

More specifically, we take the proposed NCMNet illustrated in Fig. 2 as an example. An iterative pruning strategy [27] has been chosen as the core architecture to alleviate the negative impact of outliers during network learning. $f_{\theta 1}(S) = (S_1, o_1)$ and $f_{\theta 2}(S_1) = (S_2, o_2)$ represent two sequential pruning modules with relevant parameters $\theta 1$ and $\theta 2$. $S_1 \in \mathbb{R}^{N_1 \times 4}$ and $S_2 \in \mathbb{R}^{N_2 \times 4}$, where $N > N_1 > N_2$, are two pruned correspondence sets. They are expected to be more reliable compared to S, determined by the learned logit values o_1 and o_2, for parametric model estimation. Next, o_2 is processed by an additional ResNet block [18] and an MLP layer for computing the inlier weight set w_2 as:

$$w_2 = \tanh(\mathrm{ReLU}(o_2)) \in [0, 1), \tag{1}$$

where $\tanh(\cdot)$ and $\mathrm{ReLU}(\cdot)$ denote activation functions. We then utilize S_2 and w_2 to estimate an essential matrix $\hat{E} \in \mathbb{R}^{3 \times 3}$ using the weighted eight-point algorithm [18], [27]. Finally, a label set $l \in \mathbb{R}^{N \times 1}$ of all correspondences can be obtained by the full-size verification operation. The architecture is denoted as follows:

$$\hat{E} = g(S_2, w_2), \tag{2}$$

$$l = v(\hat{E}, S), \tag{3}$$

where $g(\cdot)$ refers to the weighted eight-point algorithm [18], which offers greater robustness compared to the traditional eight-point algorithm [14] due to the consideration of inlier weights. Noteworthily, in weighted eight-point algorithm, the adopted self-adjoint eigendecomposition operation is differentiable with respect to the inlier weights, which facilitates the end-to-end regression of the essential matrix. $v(\cdot)$ is the full-size verification operation based on the epipolar constraint [14] to avoid that some inliers are removed incorrectly. Note a correspondence with the epipolar distance calculated by $\hat{E}$ less than the threshold of 10^{-4} is deemed to be an **inlier**:

$$S_{in} = \left\{ s_i \mid l_i < 10^{-4} \right\}, \tag{4}$$

where S_{in} is the retained inlier set. Similarly, **outlier** can be defined as:

$$S_{out} = \left\{ s_i \mid l_i > 10^{-4} \right\}, \tag{5}$$

where S_{out} is the outlier set. This is the same as the determination of ground-truth correspondence labels.

3.2 Enhanced Global-Graph Space

Neighbor consistency is an effective clue for discriminating correspondences, which leverages the fact that the neighbors of inliers are compatible with each other while outliers scatter randomly. Therefore, it is important to seek reliable consistent neighbors for each inliers. In this paper, we adopt a differentiable manner to leverage the potential of three types of neighbors for dealing with complex matching situations. Three different neighbor search spaces, including the coordinate space $S \in \mathbb{R}^{N \times 4}$, the feature space $F \in \mathbb{R}^{N \times d}$, and our global-graph space, are adopted to seek different types of neighbors. d is the number of channels. S

denotes the network input, and F represents the middle feature map learned from several ResNet blocks. We can obtain spatial k-nearest neighbors of each correspondence by performing knn search on the S. Similarly, feature-space k-nearest neighbors are acquired on the F. The spatial and feature-space neighbors focus on the correspondences with similar low-dimensional coordinates and high-dimensional features. Our global-graph space aims at finding globally consistent neighbors with similar graph structures through modified Graph Convolution Network [35], [27]. Compared to [36], we introduce spatial consistency of correspondences to complement original feature consistency on the global graph construction, called enhanced global-graph space.

To be specific, we firstly construct a weighted global graph $\mathcal{G}^g = \{\mathcal{V}^g, \mathcal{E}^g\}$, where nodes $\mathcal{V}^g$ denote all correspondences, and undirected edges $\mathcal{E}^g$ connect every two correspondences using the enhanced compatibility score s^c_{ij} as:

$$s^c_{ij} = s^f_{ij} \odot s^s_{ij}, 1 \leq i, j \leq N. \tag{6}$$

It indicates the affinity relationship of correspondence s_i and s_j based on the feature and spatial consistency scores. As in [36], we estimate the initial inlier weights w^p based on the F:

$$w^p = \mathrm{ReLU}(\tanh(\mathrm{MLP}(F))), \tag{7}$$

in which $\mathrm{MLP}(\cdot)$ denotes an MLP layer to reduce the channel dimension to 1. Then, the feature consistency score between the two correspondences is calculated as:

$$s^f_{ij} = w^p_i \cdot w^p_j, \tag{8}$$

which measures the degree of feature similarity among correspondences. Moreover, we additionally utilize the spatial consistency score between correspondence pairs to complement the feature consistency score as follows:

$$s^s_{ij} = \max(0, 1 - \frac{d^2_{ij}}{\epsilon^2_d}), \tag{9}$$

where $max(0, \cdot)$ operation is used for avoiding a negative value. $d_{ij} = |||u_i - u_j|| - ||v_i - v_j|||$ is the spatial difference of two correspondences, *i.e.*, $s_i = (u_i, v_i)$ and $s_j = (u_j, v_j)$, according to the length constraint. ϵ_d denotes a distance hyper-parameter for controlling the sensitivity of length constraint. Two correspondence s_i and s_j with the d_{ij} greater than ϵ_d are deemed spatially incompatible, and are received zero for s^s_{ij}. Conversely, they are spatially compatible when s^s_{ij} gives a large value, which can be used as a dependable regulator for the feature consistency score. Hence, we can formulate a weighted adjacency matrix $A = s^c_{ij} \in \mathbb{R}^{N \times N}$ of weighted global graph $\mathcal{G}^g$, which describes the long-range dependence among correspondences. A strong association is formed only when two correspondences have high feature and spatial consistency scores simultaneously, otherwise, the link will be weak or nonexistent. Finally, the spectral graph convolution operation [35] has been utilized for further learning this association:

$$L = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}, \tag{10}$$

$$F^g = \sigma(LFW^g), \tag{11}$$

in which $\tilde{A} = A + I_N$ denotes the adjacency matrix with added self-connections of the diagonal identity matrix I_N. $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ is the diagonal degree matrix of $\tilde{A}$. The graph Laplacian matrix L adjusts the F into the spectral domain. W^g represents the learned weight. $\sigma(\cdot)$ is the $\mathrm{ReLU}(\cdot)$ activation function. $F^g \in \mathbb{R}^{N \times d}$ is the enhanced global-graph space, which can effectively reflect the global consistency of correspondences from two different aspects, particularly for inliers. This could pull the nearest-neighbor distance of correspondences with similar graph structures, enabling them to become neighbors, in our enhanced global-graph space. We then perform knn search on the F^g to obtain global-graph k-nearest neighbors for each correspondence. The neighbor search region of global-graph neighbors is large (see the ablation) owing to the global operation.

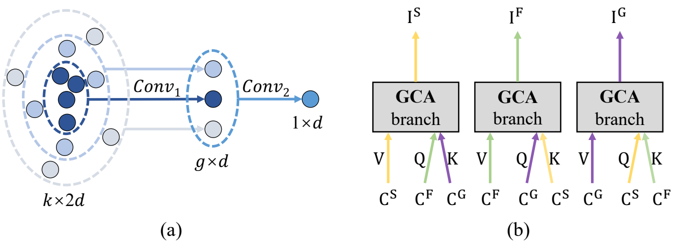

Fig. 3. (a) The proposed grouped convolution manner in our SCE layer. It divides the neighbor nodes of into g groups according to their affinities to the anchor. Two consecutive convolution layers ($Conv_1$ and $Conv_2$) are used to dynamically extract the intra-neighbor context. (b) The structure of the CCI layer. **GCA**: grouped cross-attention. It contains three parallel GCA branches to integrate inter-neighbor information. Values (V), queries (Q), and keys (K) learned from different neighbor context features are used for cross-attention operation.

3.3 Neighbor Consistency Block

Three neighbor search spaces focus on distinct types of neighbors. Thus, to enhance the robustness of method for challenging matching situations, we present a neighbor consistency (NC) block to mine the consistency of three neighbors progressively. As the core structure of NCMNet, our NC block consists of three crucial parts: neighbor embedding construction, self-context extraction (SCE) layer, and cross-context interaction (CCI) layer.

Neighbor embedding construction. When the three types of neighbors are searched, we first need to consider how to build corresponding neighbor embeddings for network learning. Graph structure is well suited for representing and modeling complex relationships between elements, making it an invaluable tool in a variety of fields [72], [73], [74], [75]. Therefore, in this component, three individual directed graphs of each correspondence s_i are built according to its different neighbors, *i.e.*, $\mathcal{G}^S_i = \{\mathcal{V}^S_i, \mathcal{E}^S_i\}$, $\mathcal{G}^F_i = \{\mathcal{V}^F_i, \mathcal{E}^F_i\}$, $\mathcal{G}^G_i = \{\mathcal{V}^G_i, \mathcal{E}^G_i\}$. Take the $\mathcal{G}^S_i$ as an example, nodes $\mathcal{V}^S_i = \{s^S_{i1}, ..., s^S_{ik}\}$ represent the spatial k-nearest neighbors of s_i, and directed edges $\mathcal{E}^S_i = \{e^S_{i1}, ..., e^S_{ik}\}$ link s_i and its spatial neighbors in $\mathcal{V}^S_i$. Following [76], [27], the edge is constructed as:

$$e^S_{ij} = [f_i, f_i - f^S_{ij}], j = 1, 2, ..., k. \tag{12}$$

f_i, f^S_{ij} denote feature maps of s_i and its j-th spatial neighbor s^S_{ij} in the $F = \{f_1, f_2, ..., f_N\}$. $f_i - f^S_{ij}$ is their residual. $[\cdot, \cdot]$ denotes the feature concatenation operation on the channel dimension. Therefore, we can obtain the spatial neighbor embedding of all correspondences $\mathcal{G}^S \in \mathbb{R}^{N \times k \times 2d}$. The feature-space neighbor embedding $\mathcal{G}^F \in \mathbb{R}^{N \times k \times 2d}$ and the global-graph neighbor embedding $\mathcal{G}^G \in \mathbb{R}^{N \times k \times 2d}$ can also be obtained using the same way.

SCE layer. After constructing the three neighbor embeddings, the next stage involves effectively mining the intra-neighbor

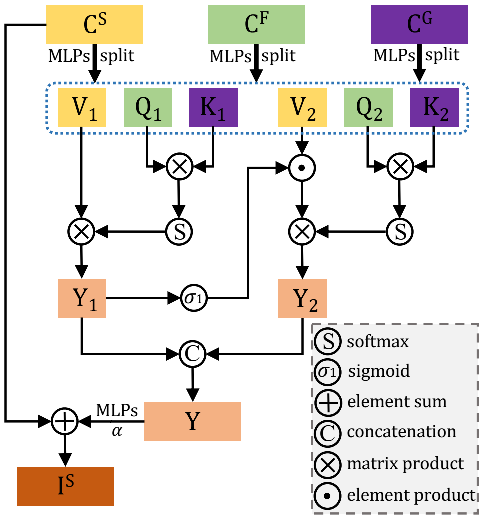

Fig. 4. The proposed grouped cross-attention (GCA) branch. The size of groups is set to 2 for ease of display. Three neighbor context features are used for generating values (V), queries (Q), and keys (K), respectively. Then, they are evenly divided into q feature groups. Besides the cross-attention operation, each feature group also receives the output of the previous group to increase the diversity and communication.

contextual information. An uncomplicated approach is to utilize well-known pooling operations, such as max-pooling and average-pooling. Nevertheless, these indiscriminate manners have the drawback of discarding the affinity relationships among graph nodes. Therefore, in order to fully leverage the graph structure of our neighbor embeddings, an SCE layer is proposed for neighbor information aggregation. Taking into account that nodes in the graph are ordered according to the similarity principle, our SCE layer employs a grouped convolution manner [27] to dynamically acquire neighbor relationships and gather neighbor context along the graph's edges.

More specifically, as shown in Fig. 3(a), given a neighbor embedding $\mathcal{G}_i \in \mathbb{R}^{k \times 2d}$ of s_i, the nodes are partitioned into g subsets according to their affinities to the anchor, with each group containing k/g nodes. The embedding is processed using two consecutive convolution layers, which are followed by a Batch Normalization (BN) [77] and the ReLU. This process is represented as follows:

$$C_i = (Conv_2(Conv_1(\mathcal{G}_i)). \tag{13}$$

$Conv_1(\cdot)$ and $Conv_2(\cdot)$ denote the convolution layers with learned $1 \times \frac{k}{g}$ kernels and $1 \times g$ kernels, respectively. For simplicity, the BN and ReLU are omitted. $C_i \in \mathbb{R}^{1 \times d}$ denotes the output of $\mathcal{G}_i$. In each NC block, three parallel SCE layers are employed to independently process each neighbor embedding, resulting in three corresponding neighbor context features denoted as $\{C^S, C^F, C^G\} \in \mathbb{R}^{N \times d}$.

CCI layer. Once the three neighbor context features are acquired, our objective is to collaboratively fuse and modulate inter-neighbor information. In [36], our CCI layer uses a cross-attention operation, which has limited capability in exploring inter-neighbor information due to the single sequential manner [62], [78], [66]. On top of that, we enrich the information integration of three features in a grouped manner. As illustrated in Fig. 3(b), the CCI layer consists of three parallel grouped cross-attention (GCA) branches. In each branch, values V are learned from one neighbor context feature, while queries Q and keys K are derived from the other two features. The overview of GCA branch is depicted in Fig. 4. We first feed three neighbor context features into an individual MLP layer followed by BN and ReLU to generate three new features $\{\mathrm{Q}, \mathrm{K}, \mathrm{V}\} \in \mathbb{R}^{N \times d}$. Then, along the channel dimension, we evenly divide them into q groups, represented by $\{\mathrm{Q}_i, \mathrm{K}_i, \mathrm{V}_i\} \in \mathbb{R}^{N \times \frac{d}{q}}$, $i \in \{1, 2, ..., q\}$. For the i-th group, the matrix multiplication between Q_i the transpose of K_i is performed, then, a softmax function is used to compute the attention weight matrix:

$$A_i^w = \text{softmax}(\mathrm{Q}_i {\mathrm{K}_i}^T), \tag{14}$$

where $A_i^w \in \mathbb{R}^{N \times N}$ measures the correlation between correspondence pairs. Next, a matrix multiplication between V_i and A_i^w is performed to enhance the V_i. Meanwhile, to boost the variety and communication between feature groups, we adopt a hierarchical multiplication operation to connect all groups. Specifically, except for the first group, other groups can utilize the output information from the preceding group. The output of each group is defined as:

$$Y_i = \begin{cases} \mathrm{V}_i A_i^w, & i = 1; \\ \sigma_1(Y_{i-1}) \odot \mathrm{V}_i A_i^w, & 1 < i \leq q, \end{cases} \tag{15}$$

where $\sigma_1(\cdot)$ is the sigmoid activation function. In this component, we employ the grouping manner to explore rich contexts from various perspectives, and the hierarchical multiplication to enhance the interaction among feature groups. These operations are beneficial for network learning owing to the combinatorial explosion effect [78], [66]. Finally, all outputs of feature groups are concatenated along the channel dimension:

$$Y = \text{concat}(Y_1, Y_2, ..., Y_i), \tag{16}$$

where Y is the final output of all groups. Here, we give the example of the first GCA branch:

$$I^S = \alpha(MLPs(Y)) + C^S, \tag{17}$$

where $MLPs(\cdot)$ consists of one MLP layer with BN and ReLU. Learned scale parameter α is initialized as 0. I^S represents the output of the first GCA branch, in which each position's response is a weighted combination between the other two neighbor features across all positions and the original features. Hence, the inliers can obtain mutual benefits within three neighbor context features by selectively aggregating contexts, which further improves the discrimination between inliers and outliers. Likewise, we can generate the results I^F and I^G by the second and third GCA branches, respectively. Three neighbor interaction features $\{I^S, I^F, I^G\} \in \mathbb{R}^{N \times d}$ constitute the ultimate outputs of the proposed NC block.

3.4 Network Architecture

With our NC block, a correspondence pruning network, called Neighbor Consistency Mining Network, is built. The specific architecture of NCMNet is illustrated in Fig. 2. It takes initial correspondences as inputs, and adopts two sequential pruning modules to progressively extract dependable candidates, which are essential for the precise prediction of parametric models and correspondence labels. Therefore, to improve the reliability of candidates, the pruning module needs to have enough capability to capture rich contexts. Each pruning module includes some off-the-shelf network structures [18], [37], [27] and our NC block for correspondence processing. As a basic structure, ResNet block [18] contains two MLP layers and several normalization techniques for correspondence learning. Order-Aware block [37] is crafted to capture both local and global contexts implicitly via an order-aware clustering operation. Global Consensus block [27] encodes global contextual information of features to estimate global scores for pruning correspondences. It's important to highlight that the feature space and our global-graph space are learned. Hence, we introduce a progressive refinement processing (employing two NC blocks within each pruning module) to enhance neighbor reliability and capture comprehensive neighbor context. Further, we use NCMNet+ to denote the NCMNet [36] using the two new improvements, $i.e.$, enhanced global-graph space and GCA branch in CCI layer.

3.5 Loss Function

Following [37], [27], the neural network is optimized by a classification loss and a regression loss:

$$\mathcal{L} = \mathcal{L}_c(o_m, y_m) + \beta \mathcal{L}_e(E, \hat{E}), \tag{18}$$

in which β represents a weight for balancing the two terms.

The classification loss $\mathcal{L}_c(\cdot)$ is a binary classification loss defined as following:

$$\mathcal{L}_c(o_m, y_m) = \sum_{m=1}^{M} H(\tau_m \odot o_m, y_m), \tag{19}$$

where M denotes the number of pruning modules. o_m is the relevant logit value of the m-th pruning module. y_m represents the weakly supervised ground-truth label of correspondence obtained by the epipolar distance d_{epi} with a default threshold of 10^{-4}. $\odot$ is the Hadamard product. $H(\cdot)$ represents the binary cross entropy function. Inliers with the epipolar distance close to d_{epi} may suffer from label ambiguity. Here, we use an adaptive temperature vector τ_m [27] to mitigate this problem:

$$\tau_i = \exp(-\frac{||d_i - d_{epi}||_1}{d_{epi}}), \tag{20}$$

where d_i is the epipolar distance of correspondence s_i. For an outlier with $d_i > d_{epi}$, the τ_i is set to 1. Thus, the inliers with a smaller epipolar distance have stronger influence for the network model optimization.

For regression loss $\mathcal{L}_e(\cdot)$, we adopt a geometry loss [57] defined as following:

$$\mathcal{L}_e(E, \hat{E}) = \frac{(p^{'T} \hat{E} p)^2}{\|Ep\|^2_{[1]} + \|Ep\|^2_{[2]} + \|E^T p'\|^2_{[1]} + \|E^T p'\|^2_{[2]}}. \tag{21}$$

Virtual correspondences (p, p') formed from the ground truth essential matrix E are used for evaluating estimated $\hat{E}$. $c_{[i]}$ represents the i-th item of vector c.

4 Experiments

In the following, we compare NCMNet/NCMNet+ with state-of-the-art correspondence pruning works. Experiments are conducted on different benchmarks to showcase the effectiveness and generalization ability of our proposed networks. The implementation details, comparative results, as well as ablation studies are presented.

4.1 Implementation Details

As shown in Fig. 2, our network takes initial correspondences generated by different feature extraction methods, including SIFT [7], ORB [9], and SuperPoint [11], as inputs. We select SIFT [7] with the nearest neighbor descriptor matching as the default technology unless otherwise specified. The number of correspondences N is about 2000, and the channel dimension d is 128 in our experiments. NCMNet utilizes the iterative pruning strategy [27] as the core structure consisting of two consecutive pruning modules with a pruning rate of 0.5. For NC block, we empirically set the neighbor number k to 9 and 6 for two pruning modules. Thus, We set the number of groups g in the SCE layer of two pruning modules to 3 and 2, respectively. The distance hyper-parameter ϵ_d in Eq. 9 is equal to 0.2. We choose the group size to be $q = 4$ in the GCA branch as [66]. The number of clusters in Order-Aware block is set as 250. **The code is provided in https://github.com/xinliu29/NCMNet to ensure the reproducibility of our results.**

Training details. We follow previous benchmark [37] to train network models implemented by Pytorch on the given datasets. Adam [79] optimizer configured with a batchsize of 32 and a learning rate of 10^{-3} has been adopted for the optimization. The training of network is conducted for 500k iterations Initially, the balance parameter β in Eq. 18 is set to 0, and then fixed at 0.5 following the first 20k iterations.

4.2 Comparative Results

We compare the proposed network against several state-of-the-art correspondence pruning works, including both conventional methods and learning-based ones. For all traditional works, we first adopt the ratio test [7] with a fixed threshold of 0.8 to remove a great number of poor initial correspondences, as they cannot deal with the case of high ratio outliers well. For all learning-based works, we utilize whole initial correspondences as inputs. We will evaluate the performance and generalization ability of methods via different tasks.

4.2.1 Geometry Estimation

We can recover the two-view geometry, including rotation and translation, by estimating essential matrix using weighted eight-point algorithm or RANSAC. The quality of geometry estimation heavily influences downstream feature matching applications. Thus, it is the main criterion for evaluating the performance of correspondence pruning algorithms.

Datasets. Following [37], we utilize Yahoo's YFCC100M [80] as the outdoor scene and SUN3D [81] dataset as the indoor scene to train and test network models. YFCC100M consists of 100 million tourist images collected from the Internet, which is split into 71 image sequences according to different landmarks, where 4 sequences are chosen for network testing. SUN3D contains a large number of image frames sampled from various RGBD videos, which is split into 254 sequences, of which 15 image sequences are for testing. The training sequences are segmented into three disjoint parts, including the training set (60%), validation set (20%), and known testing set (20%). It is important to highlight that the indoor scene is particularly difficult as it often involves numerous texture-less regions and repetitive structures.

Evaluation. The camera poses of image pairs are encoded using rotation and translation vectors calculated from the estimated essential matrix. The angular differences between ground truth

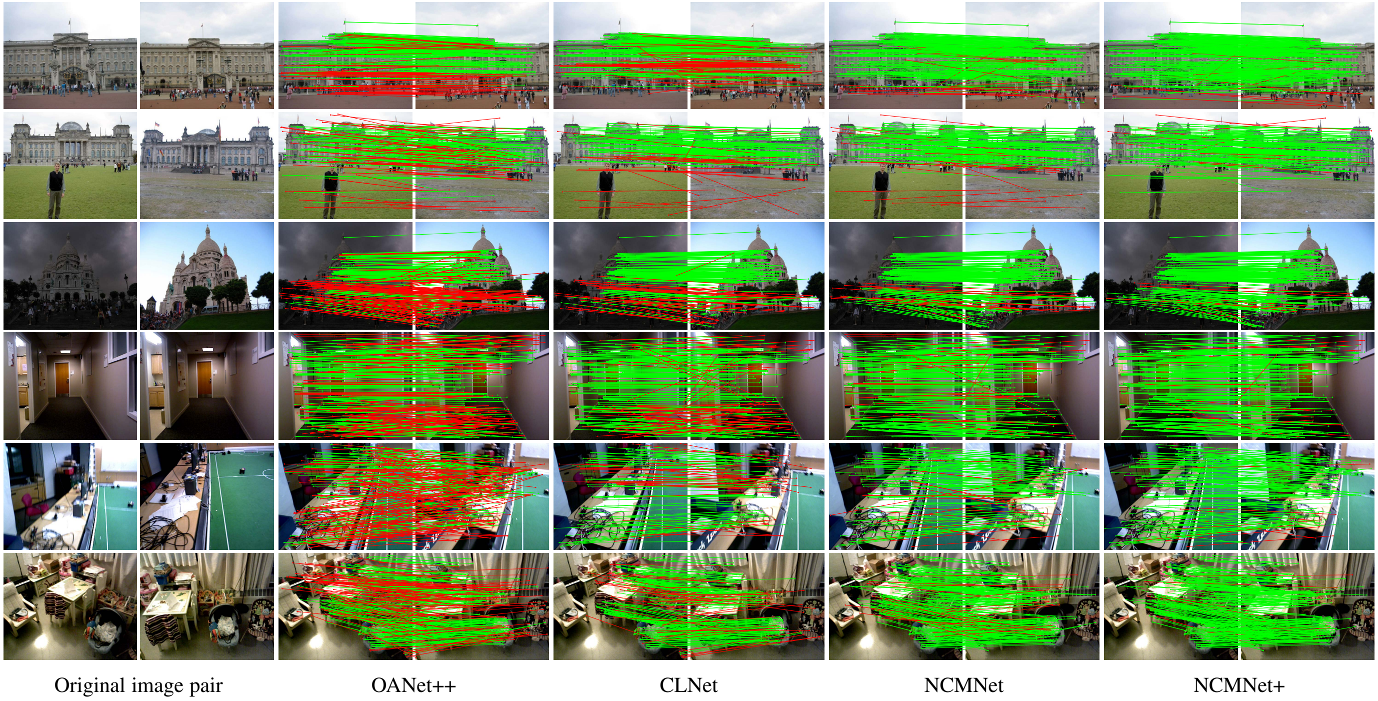

Fig. 5. Visualization results of correspondence pruning. From left to right columns: the matching images, OANet++ [37], CLNet [27], NCMNet [36], and NCMNet+. The top three examples are selected from unknown YFCC100M [80] and the rest examples are come from unknown SUN3D [81]. These image pairs involve large illumination changes, viewpoint variations, occultations, repetitive structures, textureless objects, etc. Inliers (green and outliers (red lines) lines) retained by the network models are exhibited.

TABLE 1
Quantitative comparison results on YFCC100M [80] and SUN3D [81]. mAP5° (%) on known and unknown scenes is given. Red and blue indicate the best and the second-best, respectively.

Methods	YFCC100M		SUN3D	
	Known	Unknown	Known	Unknown
RANSAC [20]	30.19	40.83	19.13	14.57
DEGENSAC [82]	21.00	27.65	16.01	11.01
GC-RANSAC [83]	30.43	41.58	18.86	14.14
MAGSAC [48]	32.80	41.61	20.35	16.24
MAGSAC++ [84]	30.48	40.95	18.90	14.19
AdaLAM [26]	32.37	45.40	21.02	15.94
LFGC [18]	16.87	25.95	11.55	09.30
DFE [57]	18.02	30.29	14.44	12.34
OANet++ [37]	33.96	38.95	20.86	16.18
ACNe [63]	29.17	33.06	18.86	14.12
LMCNet [16]	33.73	47.50	19.92	16.82
T-Net [60]	41.33	48.20	22.38	17.24
MS^{2}DG-Net [28]	39.68	48.20	22.20	17.84
MSA-Net [65]	39.53	50.65	18.64	16.86
CLNet [27]	39.16	53.10	20.35	17.03
PGFNet [66]	42.06	53.70	23.66	19.32
NCMNet [36]	52.33	63.43	26.12	20.66
NCMNet+	52.40	65.83	25.99	21.18

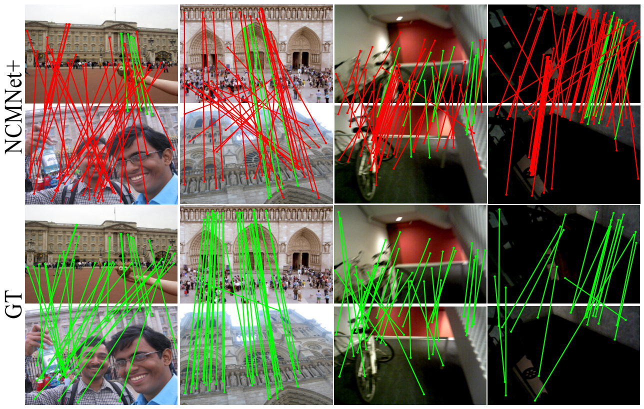

Fig. 6. Failure cases of our proposed NCMNet+ on YFCC100M [80] and SUN3D [81]. We also show the ground-truth inliers.

vectors and estimated ones are selected as the error metrics. The mean Average Precision (mAP) with different thresholds is computed as the evaluation metric of two-view geometry estimation, where mAP under 5° ($i.e.$, mAP5°) is the default metric.

Results. In Table 1, the quantitative comparison results for essential matrix estimation on YFCC100M and SUN3D datasets are reported. We see that using the ratio test can significantly enhance the RANSAC performance since most of the outliers in initial correspondences have been removed beforehand. This proves that RANSAC-related methods are difficult to cope with in the case of high outliers. Traditional methods equipped with the ratio test can deliver competitive results compared with some learned methods. Obviously, our methods are superior to all traditional and learning-based baselines by a significant margin in every testing scene. For example, compared to the second-best PGFNet [66], NCMNet obtains 10.27% and 9.73% mAP5° substantial improvements for both known and unknown outdoor scenes, respectively. The proposed NCMNet+ can further boost performance by using the enhanced global-graph space and modified inter-neighbor interaction. Fig. 5 displays the visual comparison results of our methods compared to the two other baselines [37], [27] on correspondence

TABLE 2
Performance comparisons when using SIFT [7] and SuperPoint [11] on unknown YFCC100M [80]. mAP5° **without/with** RANSAC [20] as a post-processing step is reported.

Methods	SIFT [7]		SuperPoint [11]	
	-	RANSAC	-	RANSAC
RANSAC [20]	-	40.83	-	34.38
LFGC [18]	25.95	50.00	24.25	42.57
OANet++ [37]	38.95	52.59	35.27	45.45
MS^{2}DG-Net [28]	48.20	57.15	37.38	46.48
CLNet [27]	53.10	59.13	39.19	48.15
PGFNet [66]	53.70	57.83	42.03	47.30
ANA-Net [64]	31.55	59.10	-	-
NCMNet [36]	63.43	63.33	48.20	52.20
NCMNet+	65.83	64.15	49.80	53.35

TABLE 3
Comparison results on unknown YFCC100M [80]. The initial correspondences are estimated by learning-based matchers, *i.e.*, SuperGlue [71] and LightGlue [13]. mAP5° and mAP10° are reported.

Methods	SuperGlue [71]		LightGlue [13]	
	mAP5°	mAP10°	mAP5°	mAP10°
RANSAC [20]	59.90	71.14	63.23	74.04
LFGC [18]	58.88	70.79	62.05	73.10
OANet++ [37]	60.93	71.98	63.50	74.15
MS^{2}DG-Net [28]	59.95	71.30	62.63	73.25
CLNet [27]	63.10	74.00	68.65	78.18
PGFNet [66]	60.73	71.90	62.33	73.65
NCMNet [36]	66.33	76.33	70.38	79.24
NCMNet+	68.25	77.30	71.70	79.69

TABLE 4
Generalization ability of networks for different dense matchers on unknown YFCC100M [80], including LoFTR [41] and DKM [85]. mAP5° and mAP10° are reported.

Methods	LoFTR [41]		DKM [85]	
	mAP5°	mAP10°	mAP5°	mAP10°
RANSAC [20]	68.58	77.58	74.85	82.13
LFGC [18]	64.93	74.64	73.45	81.43
OANet++ [37]	64.85	74.68	73.75	81.28
MS^{2}DG-Net [28]	66.90	76.09	73.15	80.76
CLNet [27]	69.78	78.35	75.28	82.30
PGFNet [66]	63.53	73.31	73.83	81.48
NCMNet [36]	70.25	78.55	75.55	82.51
NCMNet+	70.75	79.00	75.80	82.53

pruning. For challenging outdoor and indoor matching scenes, such as large illumination changes, viewpoint variations, occultations, repetitive structures, and textureless objects, the proposed methods obtain reliable pruning results. Besides, some failure cases are illustrated in Fig. 6, in which outliers are dominated. We can find that in these cases, the ground truth inliers are quite sparse compared to the initial correspondences (2000) due to limited overlapping regions, blurs, and low light conditions. This makes correspondence pruning more difficult. According to Table 3 and Table 4, a stronger correspondence estimator is potential to alleviate this issue. Another hidden observation is that these weakly supervised ground-truth labels may be unreliable in these cases, which will disturb the learning and optimization process of our method. We suggest that improving the robustness of the model to such noisy labels is quite valuable and meaningful in future research.

Furthermore, a robust model estimator RANSAC [20] has been utilized as a post-processing technique for learning-based methods to estimate the essential matrix. It takes the retained correspondences of networks as inputs, and adopts the inlier threshold of 0.001 as [37]. We also consider using the learned feature extraction method to detect pixel-level keypoints and construct corresponding descriptors. SuperPoint [11] proposes a self-supervised framework for keypoint detection and description, and receives widespread attention in multiple-view geometry problems. Here, we adopt SuperPoint with nearest neighbor matching strategy to construct initial correspondences of two images. The quantitative results on unknown YFCC100M dataset are reported in Table 2. For ANA-Net [64], we utilize the publicly available network model directly due to the unavailability of the training process. Again, our NCMNet and NCMNet+ outperform all baselines in all cases. The accuracy of camera poses can be further improved by applying RANSAC as a post-processing step, especially for those methods that show poor performance (*e.g.*, LFGC and OANet++) when using the weighted eight-point algorithm. All learned methods with RANSAC post-processing surpass the vanilla RANSAC, which demonstrates that learning-based correspondence pruning is more effective than simple ratio test. However, we find that the performance of NCMNet and NCMNet+ with RANSAC decreases when SIFT is used. It is difficult for RANSAC to further distill suitable inliers from the retained correspondences of networks, while the weighted eight-point algorithm can fully utilize the inliers and their weights. Meanwhile, we find that the methods using SuperPoint perform worse than the ones using SIFT. This is attributed to the learned feature extraction method having limited generalization ability, so the quality of estimated initial correspondences solely using nearest neighbor search is difficult to guarantee.

Learning-based Matchers. Recently, SuperGlue [71] takes keypoints produced by feature methods as inputs, and leverages graph neural networks to improve keypoint discrimination and learn optimal assignments. It can be viewed as a replacement for nearest neighbor matching of descriptors. Here, we first use SuperPoint [11] as the keypoint detector. Then, more advanced SuperGlue and LightGlue [13], which is an effective and efficient replacement of SuperGlue, as the keypoints matchers are selected to estimate initial correspondences, where network models are provided by the authors. Finally, we adopt learned correspondence pruning methods retrained on the datasets generated by SuperGlue with RANSAC to estimate the essential matrix. Moreover, we find the epipolar distance threshold in the full-size verification step has a significant impact on the performance, so we set it to $1e$-7 in this experiment. In Table 3, we give the comparison results on unknown YFCC100M dataset. We can see that RANSAC can achieve significant performance gains. This is because learning-based SuperGlue and LightGlue can generate more accurate initial correspondences by improving the representation ability of keypoints. For example, the mAP5° is raised from 34.38% as reported in Table 2 to 59.90% when using SuperGlue. As a result, many learning-based pruning methods are difficult to further increase

TABLE 5
Generalization ability of networks on the outdoor PhotoTourism and Pragueparks [46] datasets, and the indoor ScanNet [86] dataset. mAP5° without/with RANSAC post-processing is reported.

Methods	PhotoTourism	Pragueparks	ScanNet
LFGC [18]	13.62/43.13	02.42/49.51	02.07/11.93
OANet++ [37]	30.35/48.39	07.37/49.17	04.73/14.27
MS^{2}DG-Net [28]	36.79/52.52	09.68/57.54	04.93/15.33
CLNet [27]	38.43/51.49	17.27/59.52	06.53/15.93
PGFNet [66]	41.22/52.34	09.90/56.00	04.87/14.93
NCMNet [36]	52.62/56.54	24.09/63.15	09.00/16.87
NCMNet+	52.93/56.84	27.17/63.48	10.13/17.47

TABLE 6
Results of fundamental matrix estimation on unknown YFCC100M [80]. mAP with different error thresholds is reported.

Methods	mAP5°	mAP10°	mAP15°	mAP20°
LFGC [18]	19.90	30.96	39.11	45.68
OANet++ [37]	30.95	42.63	50.79	56.82
MS^{2}DG-Net [28]	36.63	48.56	56.68	62.76
CLNet [27]	47.98	57.51	63.88	68.50
PGFNet [66]	40.20	51.13	58.35	63.54
NCMNet [36]	53.03	62.89	68.83	73.18
NCMNet+	54.60	63.79	69.53	73.70

the accuracy of recovered camera poses. In comparison, our NCMNet and NCMNet+ are able to obtain decent performance gains in both cases since our pruning methods can provide more suitable inliers for geometry estimation. This further demonstrates the compatibility of our methods with more advanced keypoint matchers [71], [13].

Generalizability. In this section, we adopt different matchers and datasets to evaluate the generalization ability of network models. Lately, some dense matchers [41], [87], [88], [85] take pairs of images as inputs to establish pixel-wise dense matches without the need for keypoints. Here, we construct the initial correspondences with state-of-the-art matchers, including semi-dense LoFTR [41] and dense DKM [85], where models are publicly available. The quantitative results on unknown YFCC100M dataset are reported in Table 4. We utilize the pruning network models trained on SuperGlue and empirically set the epipolar distance threshold to $1e$-2. Compared to sparse matchers, these dense approaches can achieve better performance since they are not limited to keypoint detectors. Our methods still bring decent improvements, indicating the usability and potential of correspondence pruning as a complementary module.

Meanwhile, we also analyze the generalizability of network models on different datasets, including the outdoor PhotoTourism and Pragueparks [46] datasets, and the indoor ScanNet [86] dataset. PhotoTourism is a photo-tourism data containing 9 scenes for testing, Pragueparks is a small-scale video sequence including 3 scenes for testing, which come from the Image Matching Challenge [46]. ScanNet is a large RGB-D video dataset, in which 1500 test pairs are provided by [71]. We adopt SIFT [7] with nearest neighbor matching to establish correspondences. Network models are trained on the outdoor YFCC100M or indoor SUN3D datasets using SIFT feature. As shown in Table 5, NCMNet and NCMNet+ exhibit better generalization ability in different matching situations than other works, further demonstrating the robustness of our methods.

TABLE 7
Results of homography matrix estimation on HEB [89]. The mean Average Accuracy of the Re-Projection Error (RPE), the Angular Pose Error (APE), the Rotation Error (RE), and the Absolute Translation Error (ATE) are reported.

Method	RPE	APE	RE	ATE
RANSAC [20]	27.89	2.71	19.25	27.34
LFGC [18]	39.02	3.74	25.47	31.06
OANet++ [37]	39.65	3.98	26.41	31.55
MS^{2}DG-Net [28]	39.22	2.96	26.01	30.59
CLNet [27]	40.08	3.98	26.44	31.56
PGFNet [66]	38.34	3.84	25.53	31.00
NCMNet [36]	42.14	4.35	27.38	32.17
NCMNet+	43.86	4.51	28.32	32.64

Fundamental Matrix Estimation. In the preceding experiments, we estimate essential matrix to obtain relative poses, which assumes that camera intrinsics are known. Estimating fundamental matrix is a more widespread way in the structure from motion (SfM) pipelines [2]. The main difference between them is that the latter employs raw image coordinates as inputs instead of normalized coordinates. Therefore, we retrain network models to estimate fundamental matrix employing weighted eight-point algorithm on the YFCC100M dataset, and use the same evaluation metrics as essential matrix estimation. As reported in Table 6, NCMNet and NCMNet+ continue to show substantial superiority over other state-of-the-art works in terms of mAP across various error thresholds. Moreover, we can observe that our methods can achieve comparable or superior camera pose estimation results recovered from fundamental matrix compared to other competitors using essential matrix.

4.2.2 Homography Estimation

The homography, which can characterize the transformation between two planes or viewpoints, plays a pivotal role in computer vision applications [14]. Finding reliable homographies of image pairs also requires accurate feature correspondences. In this experiment, we evaluate the correspondence pruning methods for homography estimation.

Dataset. HEB [89] is a large-scale homography dataset, which contains 226,260 homographies between image pairs sampled from the Pi3D dataset [89]. The testing set consists of nine scenes with significant changes for the viewpoint and illumination, thus making the low inlier ratio. Considering that the training data is not enough, we follow the evaluation strategy in [89], where network models pre-trained on the YFCC100M dataset are used for evaluation.

Evaluation. Following benchmark [89], we present the mean Average Accuracy of the Re-Projection Error (RPE), the Angular Pose Error (APE), the Rotation Error (RE), and the Absolute Translation Error (ATE) to assess how models perform in filtering out outliers for homography matrix estimation. The mAA of four kinds of error is calculated with the following thresholds: from 1 to 10 degrees for APE and RE, from 0.1 to 5 meters for ATE, and from 1 to 20 pixels for RPE.

Results. The quantitative comparison results of correspondence pruning methods for homography estimation are tabulated

TABLE 8
Ablation studies regarding performance gains of the key components in each pruning module. **IPS**: the iterative pruning strategy. **SCE**: the Self-Context Extraction layer. **CCI**: the Cross-Context Interaction layer. **PNR**: the progressive neighbor refinement processing. **OA**: the Order-Aware block. **Bold** indicates the best.

IPS	SCE	CCI	PNR	OA	mAP5°	mAP20°
✓					53.10	76.11
✓	✓				56.50	78.34
✓	✓	✓			58.63	80.03
✓	✓	✓	✓		61.73	81.46
✓	✓	✓	✓	✓	**63.43**	**82.46**

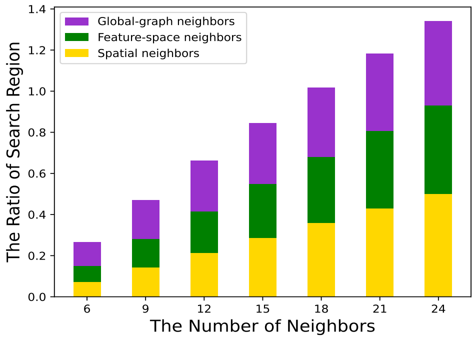

Fig. 7. Illustration of mean neighbor search region ratio (%) for all inliers in terms of different neighbor numbers of k. Global-graph space can find neighbors at greater distances due to the consideration of long-range dependencies.

in Table 7. The initial correspondences are provided by [89]. For learning-based models, they are adopted for filtering out outliers while RANSAC [20] estimator is used for the final homography matrix estimation. It can be seen that our methods outperform all traditional and learning-based methods in all cases. For example, compared to the second-best CLNet, NCMNet obtains 2.06%, 0.37%, 0.94%, and 0.61% mAA improvements on four metrics, respectively. Meanwhile, the experimental results indicate that our NCMNet+ can further improve the performance.

4.3 Ablation Studies

In this section, we construct ablation studies to examine the Performance of different components in the proposed NCMNet on the unknown YFCC100M [80] dataset. We use both mAP5° and mAP20° as metrics to assess the methods.

Main components. In the proposed NCMNet, the iterative pruning strategy [27] is adopted as the network framework. To validate the effect of the main components in the pruning module, we evaluate their performance gains compared to the baseline [27]. The SCE layer is utilized to extract the intra-neighbor context, meanwhile the CCI layer aims at exploring the inter-neighbor interaction. To boost the reliability of dynamic neighbors, we use the progressive neighbor refinement processing. While the Order-Aware block helps in implicitly obtaining local and global contexts. The performance improvements of the main components in each pruning module are tabulated in Table 8. It is evident that the model performance gradually improves with the incremental addition of SCE layer and CCI layer. Besides, from the results of the 4-th and 5th rows, it can be demonstrated that adopting the progressive neighbor refinement processing and the Order-Aware block is effective. Our NCMNet (IPS + SCE + CCI + PNR +OA) can achieve the best performance, which verifies the effectiveness and reasonability of each main component.

TABLE 9
The effectiveness of simultaneously using three types of neighbors. **SN**: the spatial neighbors. **FN**: the feature-space neighbors. **GN**: the global-graph neighbors.

	Three SN	Three FN	Three GN	SN+FN+GN
mAP5°	61.40	62.60	61.73	**63.43**
mAP20°	81.26	81.74	81.31	**82.46**

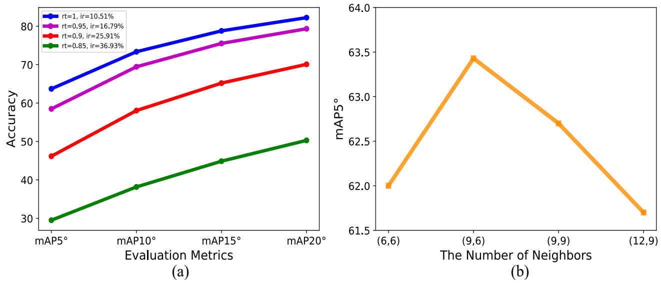

Fig. 8. (a) Influence of different inlier ratios in initial correspondences. The ratio test (rt) equipped with different thresholds is utilized to obtain the input sets with different inlier ratios. We report mAP with different error thresholds. (b) Parametric analysis of different numbers of neighbors in neighbor embeddings. $(\cdot,\cdot)$ represents the neighbor amount k in each neighbor embedding of the first and the second pruning modules, respectively. mAP5° is reported.

Three types of neighbors. In Fig. 1 (c), we give a visual comparison about three types of neighbors. To further show the search region of three neighbors, Fig. 7 presents quantitative results on the mean neighbor search region for all inliers. It is the average ratio between the area of the rectangle covered by all neighbors and the whole image. Here, owing to the consideration of long-range dependencies among correspondences, the global-graph neighbors of inliers exhibit larger neighbor search regions compared to the others for different numbers of k-nearest neighbors. In addition, to demonstrate the complementary among the three types of neighbors, we employ the same neighbor embeddings within the NC block for evaluation. The comparative results are tabulated in Table 9. It is evident that when the three types of neighbors are all used, the network achieves the best performance.

Neighbor context aggregation. To dynamically extract the neighbor context for each neighbor embedding, our SCE layer utilizes a grouped convolution method. Thus, we compare it with some classic aggregation manners, including the average-pooling layer, the max-pooling layer, and the convolution layer with $1 \times k$ kernels, to prove the validity of this design. The comparative results are reported in Table 10, where the grouped convolution strategy surpasses other competitors, indicating its effectiveness.

Inlier ratio of inputs. The performance of traditional methods, *e.g.*, RANSAC [20] and its variants [82], [83], [48], is highly dependent on the inlier ratio (ir) in the initial correspondences. Therefore, we analyze the influence of the inlier ratio for NCMNet as illustrated in Fig. 8 (a). Using Lowe's ratio test (rt) [7] with different thresholds during descriptor matching, we construct initial correspondences with varying inlier ratios as network inputs,

TABLE 10
Quantitative comparisons of different context aggregation manners in the SCE layer. **"Avg-pooling & MLPs"** aggregates neighbor context with an average-pooling layer and two successive MLP layers with BN and ReLU. **"Max-pooling"** denotes a max-pooling layer.

	mAP5°	mAP20°
Avg-pooling & MLPs	61.48	81.53
Max-pooling & MLPs	62.75	81.86
$1 \times k$ kernels Conv.	62.88	81.91
Grouped Conv.	**63.43**	**82.46**

TABLE 11
The different combination strategies of three types of neighbors. **SN**: the spatial neighbors. **FN**: the feature-space neighbors. **GN**: the global-graph neighbors. **CCI**: the Cross-Context Interaction layer. **RT**(ms): the average runtime. **FLOPs**(G): the floating point operations per second.

baseline	SN	FN	GN	CCI	mAP5°	mAP20°	RT	FLOPs
✓					25.95	54.63	**5.01**	**0.86**
✓	✓				30.17	58.13	5.77	1.55
✓		✓			31.15	59.66	5.83	1.55
✓			✓		30.73	60.97	10.18	1.58
✓	✓	✓			33.22	61.66	6.51	2.31
✓	✓		✓		33.70	62.98	11.00	2.34
✓		✓	✓		33.83	62.60	10.89	2.34
✓	✓	✓	✓		36.03	63.96	11.63	3.10
✓	✓	✓	✓	✓	**37.83**	**65.94**	13.90	3.35

in which NCMNet is retrained under corresponding training sets. In contrast to conventional approaches, our method demonstrates effectiveness even with low inlier ratios. Although the ratio test proves beneficial in reducing outliers of inputs, it also makes many important inliers discarded, which will diminish overall performance. The results also demonstrate that our network is more robust under challenging conditions, *i.e.*, there are sufficient inliers but many outliers in the initial correspondences.

TABLE 12
The effectiveness of the two proposed improvements. **EGS**: the enhanced global-graph space. **GCA**: the grouped cross-attention branch.

	baseline	EGS	GCA	EGS+GCA
mAP5°	63.43	64.85	64.40	**65.83**
mAP20°	82.46	82.60	82.68	**83.14**

Combinations of Neighbors. In this work, we propose to find three types of neighbors for each correspondence for adapting complex matching situations. Here, we verify the effectiveness and efficiency of this design by comparing different combinations of three types of neighbors. We select LFGC [18] as the comparative baseline, which contains 12 sequential ResNet blocks. We insert the different combinations of neighbor embeddings and the CCI layer into the middle of the baseline for mining different neighbor consistency information. For two or three types of neighbor context features, we directly adopt the concatenation operation along the channel dimension to fuse their information. Table 11 reports the performance gains and computational overhead. It can be seen that better performance improvements are obtained when using either one or two types of neighbors. When employing three types of neighbors simultaneously, we can acquire the best performance gains compared with the baseline, which further demonstrates the three types of neighbors are complementary. However, we find the construction of global-graph neighbor requires more runtime compared to the other two neighbors. This is because GCN operation needs a high computational cost, especially for more input correspondences. With above observations, a more efficient GCN strategy, such as the graph clustering [90], topology sampling [91] and pipeline parallelism [92], is necessary for some real-time feature matching applications. In the future, how to improve the efficiency of global-graph construction is our main focus.

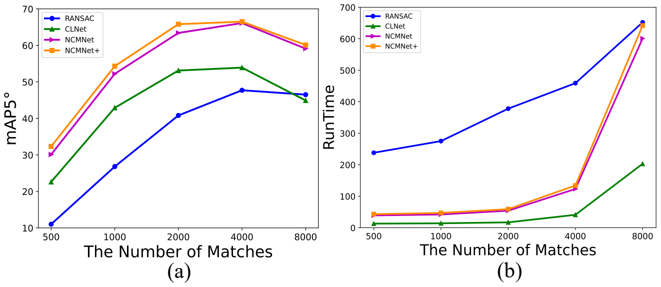

Fig. 9. The influence of different numbers of input matches. SIFT and nearest neighbor search are adopted to establish 500/1000/2000/4000/8000 matches. (a) The average runtime (ms) and (b) mAP5° (%) are shown.

Parametric Analysis of k. In the NC block, we seek k-nearest neighbors for each correspondence in different neighbor search spaces to construct neighbor embeddings. A suitable neighbor number k is extremely important for extracting neighbor context. Fig. 8 (b) gives the results of different combinations of neighbor number k. Our NCMNet with the combination of $k = (9, 6)$ attains the best performance over other settings. Therefore, the k in the two pruning modules is set to 9 and 6, respectively.

Effectiveness of improvements. Based on the previous version [36], we propose an enhanced global-graph space and a grouped cross-attention branch in the CCI layer. The former utilizes the spatial consistency inherent in correspondences to complement the feature consistency used in [36]. It aims to construct a global connected graph by the affinity relationship between correspondences from spatial and feature aspects to enhance the long-range dependencies. The latter explores rich inter-neighbor information interaction based on an effective hierarchical grouped manner to replace the single cross-attention operation in [36]. We select NCMNet [36] as the baseline, and then add the two proposed operations to the baseline. As shown in Table 12, when equipped with the enhanced global-graph space or the grouped cross-attention branch, the baseline's performance is further improved. Enhanced global-graph space is able to provide more reliable globally consistent neighbors. Grouped cross-attention branch can enrich the information integration of three neighbor features. The proposed NCMNet+ delivers the best performance, which demonstrates the effectiveness of two improvements.

Different number of inputs. The number of input correspondences may change with different estimation methods, which will affect the effectiveness and efficiency of pruning works. So we test the effect of different numbers of correspondences estimated by SIFT and nearest neighbor search in terms of performance and runtime. The pruning models are trained using 2000 correspondences. As illustrated in Fig. 9, as the number of

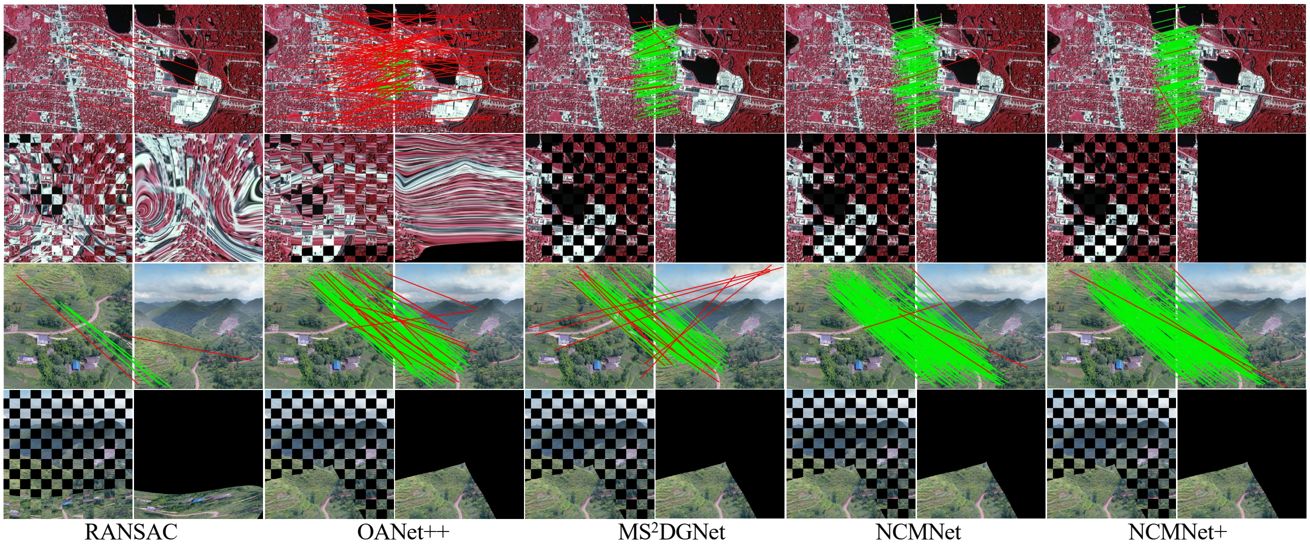

Fig. 10. Visualization results of correspondence pruning and image registration. From left to right columns: RANSAC [20], OANet++ [37], MS^{2}DG-Net [28], NCMNet [36], and NCMNet+. In the 1st and 3rd rows, red lines denote outliers and green lines denote inliers retained by methods. The checkerboard images and warped images are shown respectively on the left and right parts in the 2nd and 4th rows.

inputs increases, the networks might obtain better performance due to the increase of potential inliers. Meanwhile, the runtime also increases dramatically due to the expensive computational overhead of multiple GCN operations. Nevertheless, as mentioned in the above ablation, we believe that this problem can be further mitigated by using more efficient GCN strategies [90], [91], [92] in the future.

5 Extended Tasks

Accurate feature correspondences between two images are important preconditions for a variety of feature matching based tasks. In this section, we will extend our methods to several important feature-based tasks, including remote sensing image registration, point cloud registration, 3D reconstruction, and visual localization.

5.1 Remote Sensing Image Registration

Image registration aims at estimating the geometric transformation and then aligning the overlapping region between source and target images. Remote sensing image registration [93] is a crucial process for some tasks ($e.g.$, image fusion, multispectral classification, and change detection), which also needs accurate feature correspondences [94], [95]. Correspondence pruning can provide reliable inliers for accurate image registration.

Datasets. We select 57 low-altitude remote sensing image pairs with various types and scenes provided by [96], [97]. They provide initial correspondences generated by using SIFT feature [7], where inliers and outliers are manually labeled. These image pairs involve large viewpoint changes as well as extreme patterns. Hence, encountering a high number of outliers is unavoidable.

Evaluation. Due to the lack of sufficient data to train networks, we directly adopt network models trained on YFCC100M dataset for testing the generalization ability. We use the root mean square error ($RMSE$), median error (MEE), as well as maximum error (MAE) as evaluation metrics to measure the registration performance of methods. These metrics are formulated as follows:

TABLE 13
Quantitative registration results on remote sensing image pairs. The average $RMSE$, MAE, MEE and runtime (RT) are used for evaluation. ↓ means that a lower value is better.

Method	$RMSE$↓	MEE↓	MAE↓	$RT(ms)$↓
RANSAC [20]	50.60	55.94	164.29	291.69
LFGC [18]	10.40	8.80	43.68	45.08
OANet++ [37]	7.17	8.79	34.10	73.80
MS^{2}DG-Net [28]	6.72	5.11	42.48	72.84
CLNet [27]	10.90	11.07	47.87	73.76
NCMNet [36]	1.55	0.01	23.88	73.47
NCMNet+	1.39	0.01	23.03	72.83

$$RMSE = \sqrt{\frac{1}{L}\sum_{i=1}^{L}\|r_i - \mathcal{F}(s_i)\|_2^2}, \tag{22}$$

$$MEE = \text{median}\left\{\|r_i - \mathcal{F}(s_i)\|_2\right\}_{i=1}^{L}, \tag{23}$$

$$MAE = \max\left\{\|r_i - \mathcal{F}(s_i)\|_2\right\}_{i=1}^{L}, \tag{24}$$

where r_i and s_i represent the keypoint coordinates of the reference image and the sensed image, respectively. $\mathcal{F}$ is the transformation function estimated by methods between two matching images. L refers to the count of keypoint coordinates. $\|\cdot\|_2$ denotes the Euclidean norm of vectors. $\text{median}(\cdot)$ and $\max(\cdot)$ calculate the corresponding median and maximal values, respectively.

Results. Table 13 gives the quantitative results with some correspondence pruning methods, where runtime indicates the registration time after obtaining refined correspondences. RANSAC with 1k iterations is used for comparison. For learning-based methods, we first utilize network models to filter outliers, and then adopt RANSAC with 50 iterations to further process retained correspondences. Compared with these competitors, our

TABLE 14
Comparative results of point cloud registration on 3DMatch [98]. The average RR, IP and IR are used for evaluation.

Descriptor	FCGF			FPFH		
Method	RR	IP	IR	RR	IP	IR
RANSAC [20]	86.57	76.86	77.45	40.05	51.52	34.31
LFGC [18]	91.56	77.49	80.85	73.66	64.60	58.67
OANet++ [37]	91.87	77.76	80.49	72.51	62.62	55.96
MS^{2}DG-Net [28]	92.05	77.76	84.35	78.54	67.99	72.10
CLNet [27]	91.81	77.99	82.72	77.94	68.26	67.12
NCMNet [36]	92.54	78.28	84.70	78.60	69.64	70.13
NCMNet+	92.98	78.69	85.92	79.71	70.19	72.92

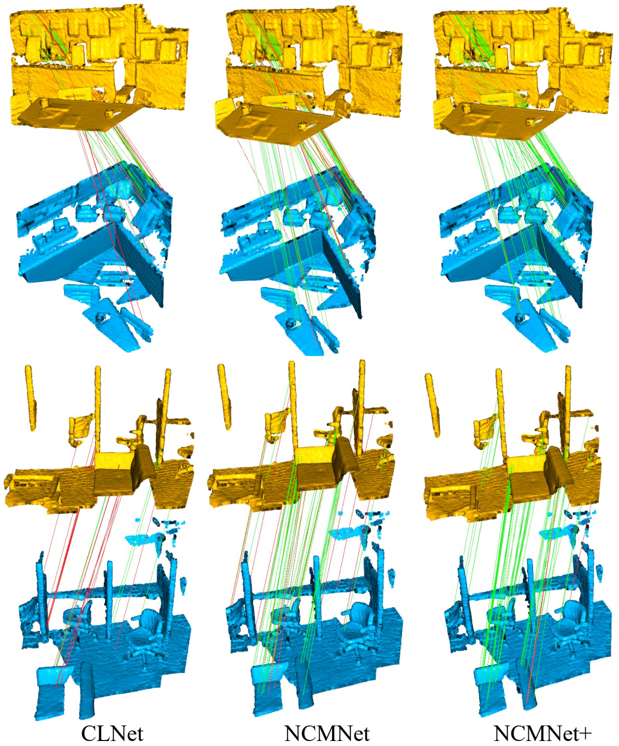

Fig. 11. Visualization results of correspondence pruning on 3DMatch[98]. From left to right columns: CLNet [27], NCMNet [36], and NCMNet+. Green and red lines represent the recognized inliers and outliers, respectively.

NCMNet and NCMNet+ show superior performance. NCMNet+ achieves the most optimal outcomes in terms of $RMSE$, MAE, and MEE. Moreover, visualization results for correspondence pruning and image registration on two typical image pairs are illustrated in Fig. 10. We see that the proposed methods can obtain more accurate results and preserve more inliers compared to other works.

5.2 Point Cloud Registration

Point cloud registration [99], [100], [101] needs to determine the optimal pose transformation for aligning a pair of point clouds, which is a critical problem in point cloud processing. Similar to image feature matching, it can be solved by establishing reliable point-to-point feature correspondences, where outliers are inevitable due to the limitation of 3D feature extraction methods and the limited overlaps. Therefore, correspondence pruning is one of the indispensable steps for handling 3D correspondences with numerous outliers.

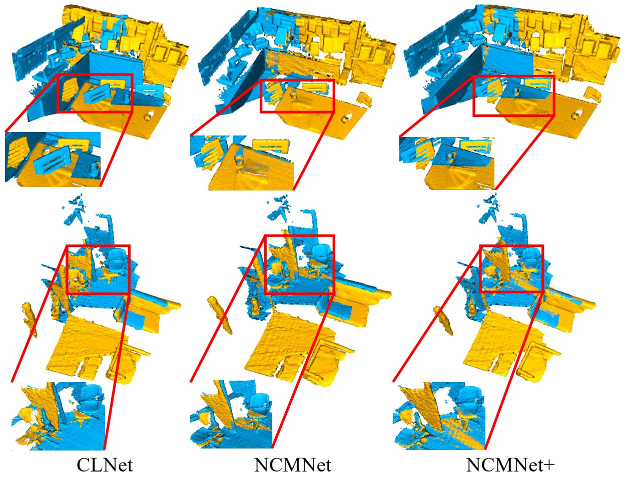

Fig. 12. Visualization results of point cloud registration on 3DMatch[98]. From left to right columns: CLNet [27], NCMNet [36], and NCMNet+.

Datasets. Owing to the different dimensions of correspondences, we use the indoor 3DMatch [98] dataset to retrain and test the network models. In the test set, there are $1,623$ point cloud fragments with partial overlap derived from eight distinct scenes.

Evaluation. We use both the learned fully convolutional geometric features (FCGF) [102] and the traditional fast point feature histograms (FPFH) [103] as feature extraction methods to construct initial correspondences. Following the training setting [104], networks are trained with 50 epochs on FCGF and then tested on both FCGF and FPFH. We adopt the registration recall (RR), the most pivotal criterion in point cloud registration, to evaluate the performance. It indicates the proportion of successful alignments, where the rotation error must be below 30cm and the translation error must be less than 20°. Meanwhile, Inlier Precision (IP: the ratio between identified inliers and retained correspondences) and Inlier Recall (IR: the ratio between identified inliers and actual inliers) are used for evaluating the performance of correspondence pruning.

Results. Quantitative comparative results on two settings are shown in Table 14. Here, RANSAC uses 1k iterations for comparison. We see that our NCMNet+ is able to obtain the best IP and IR on the two settings, which demonstrates the outstanding correspondence pruning performance. For RR, the proposed NCMNet+ outperforms all comparative methods due to the better correspondence results. Furthermore, Fig. 11 and Fig. 12 give two visualization results of some methods on both correspondence pruning and registration, respectively, which further demonstrates the effectiveness of our methods.

5.3 3D Reconstruction

The process of 3D reconstruction entails recovering a 3D model of an object or scene using several 2D images [105]. Usually, the performance of reconstruction relies greatly on the quality of correspondences in 2D image pairs. Therefore, we evaluate the generalization of networks for the 3D reconstruction task.

TABLE 15
Quantitative comparative results of 3D reconstruction on the different-scale datasets. R+M: ratio test + mutual check

Dataset	Method	Reg	Sparse	Dense	TL	Obs	Reproj↓
Fountain	OANet++ [37]	11	12456	306598	4.88	5535	0.45*px*
	MS^{2}DG-Net [28]	11	10584	313808	4.86	4681	0.45*px*
	CLNet [27]	11	12029	320722	4.87	5326	0.46*px*
	NCMNet [36]	11	12319	327812	4.85	5609	0.46*px*
	NCMNet+	11	14354	373133	5.01	6331	0.43*px*
Herzjesu	OANet++ [37]	8	7406	278702	4.24	3944	0.49*px*
	MS^{2}DG-Net [28]	8	7229	254100	4.22	3815	0.48*px*
	CLNet [27]	8	7498	245425	4.25	3981	0.49*px*
	NCMNet [36]	8	7539	261289	4.27	3998	0.50*px*
	NCMNet+	8	7773	282455	4.31	4146	0.49*px*
South-Building	OANet++ [37]	126	128245	2350753	5.36	5456	0.58*px*
	MS^{2}DG-Net [28]	127	110953	2297649	5.65	4940	0.57*px*
	CLNet [27]	127	119784	2195264	5.53	5216	0.58*px*
	NCMNet [36]	128	136360	2199067	5.26	5608	0.59*px*
	NCMNet+	128	121082	2386528	5.68	5277	0.58*px*
Gendarmenmarkt	OANet++ [37]	958	334503	1512576	5.56	1942	0.79*px*
	MS^{2}DG-Net [28]	976	324324	1358179	5.73	1882	0.79*px*
	CLNet [27]	973	333665	1238846	5.71	1950	0.78*px*
	NCMNet [36]	970	349821	1391622	5.45	2057	0.80*px*
	NCMNet+	1006	368701	1635901	5.71	2001	0.78*px*
Alamo	OANet++ [37]	806	219696	1701979	11.34	3020	0.72*px*
	MS^{2}DG-Net [28]	858	220835	2036959	11.29	2907	0.72*px*
	CLNet [27]	859	221539	2954842	11.13	2857	0.70*px*
	NCMNet [36]	838	240389	2833031	10.61	3034	0.74*px*
	NCMNet+	865	257033	3147980	11.68	3064	0.72*px*

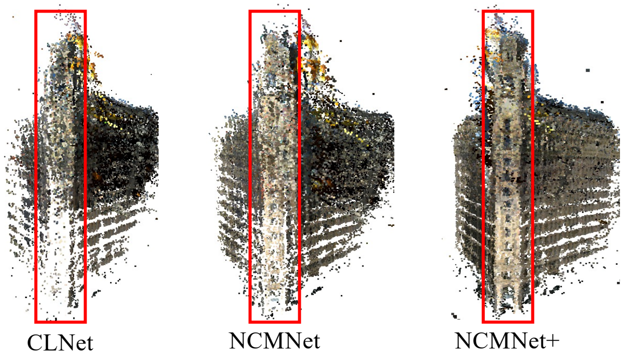

Fig. 13. 3D reconstruction visualization results of the Alamo dataset. From left to right: the results of CLNet [27], NCMNet [36], and NCMNet+.

Datasets. Following [106], we conduct a sequence of 3D reconstruction experiments utilizing the ***COLMAP*** [2] platform. The testing datasets contain some small and medium scale subsets from YFCC100M [80]. Specifically, the small scale subsets, including Fountain (11 images), Herzjesu (8 images) and South-Building (128 images), utilize the enumeration manner to choose similar images. As [107], the Bag of Word (BoW) model is selected to search for the top 20 similar images on the medium scale subsets, containing Gendarmenmarkt (1,463 images) and Alamo (2,915 images).

Evaluation. All the learning-based models are trained using the YFCC100M dataset with $4,000$ keypoints extracted by SIFT. We report registered images number (Reg), sparse points number ($Sparse$), dense points number ($Dense$), mean track length (TL), and reprojection error ($Reproj$) to evaluate how models perform in improving the quality of correspondences for 3D reconstruction. Especially, $Sparse$ and $Dense$ are the main indicators for evaluation.

TABLE 16
Quantitative comparative results of visual localization on the Aachen Day-Night v1.1 dataset [109], [110]. The percentage of correctly localized queries at different thresholds is reported.

Methods	Day	Night
	(0.25m, 2°) / (0.5m, 5°) /(5m, 10°)	
SIFT [7]	65.3 / 72.0 / 78.9	16.8 / 19.9 / 27.7
LFGC [18]	77.4 / 82.8 / 86.9	30.9 / 34.6 / 41.4
OANet++ [37]	79.1 / 84.8 / 89.0	33.0 / 37.2 / 45.5
MS^{2}DG-Net [28]	76.8 / 83.3 / 86.9	26.2 / 31.4 / 42.4
CLNet [27]	82.8 / 89.4 / 93.4	39.8 / 50.8 / 61.8
NCMNet [36]	82.8 / 91.1 / 95.0	43.5 / 53.9 / 69.1
NCMNet+	84.2 / 92.5 / 96.0	48.2 / 59.7 / 75.4

Results. The quantitative comparison results for 3D reconstruction are tabulated in Table 15. These learning-based correspondence pruning methods are adopted to improve the quality of correspondences. Our methods consistently yield better performance when compared with the other works. Meanwhile, as shown in Fig. 13, the visualization results on the Alamo further indicate the validity of our methods. Our NCMNet+ gives more complete reconstruction results compared to the other two methods, especially in the red box.

5.4 Visual Localization

Visual localization aims at estimating the 6 Degree-of-Freedom (DoF) camera pose of a query image with respect to a visual representation of reference scene, such as a 3D scene model [6], [108]. The process of 3D structure-based visual localization needs employing local features to generate 2D-3D correspondences for pose estimation. Correspondence pruning is necessary for further identifying adequate and accurate correspondences, which is crucial for successful visual localization.

Datasets. We perform the experiment on the Aachen Day-Night v1.1 dataset [109], [110] which focuses on the localization under severe illumination changes. The dataset includes 6,697 daytime reference images, 824 daytime query images, and 191 nighttime query images taken from mobile devices. We evaluate performance on the Long-Term Visual Localization benchmark [111].

Evaluation. The correspondence results obtained by network models are integrated into the open-sourced localization pipeline HLoc [112]. Specifically, we construct 2,000 initial correspondences with SIFT between query and reference images, and then use correspondence pruning methods to obtain reliable correspondences. We utilize ***COLMAP*** [2] to triangulate a reference 3D SfM model and recover their poses. The network models are trained on the YFCC100M datasets with 2,000 SIFT keypoints. We use the proportion of correctly localized queries at various thresholds as the evaluation metric.

Results. The quantitative comparative results for visual localization are shown in Table 16. We can see that our NCMNet+ outperforms all baselines by a significant margin in both daytime and nighttime scenes at different thresholds. For severe illumination changes, our methods are able to deliver superior results than competitors, demonstrating their superiority and robustness for challenging visual localization.

6 Conclusion

In this work, we propose an effective architecture, called Neighbor Consistency Mining Network (NCMNet), for challenging correspondence pruning. A global-graph space has been developed to search for consistent neighbors by explicitly modeling the long-range affinity relationship between correspondences with a global connected graph. Meanwhile, we design a neighbor consistency block, which progressively mines the consistency of three types of neighbors, for enhancing the robustness of method. We also introduce spatial consistency to enhance the reliability of global-graph space and hierarchical grouped manner to enrich the inter-neighbor information integration. Comprehensive experiments on different benchmarks and extended tasks have been conducted to validate the effectiveness and generalization ability of NCMNet and NCMNet+, demonstrating clear superiority over the state-of-the-arts.

Acknowledgments

This work was supported by Natural Science Foundation of Tianjin, China (No. 20JCJQJC00020), the National Natural Science Foundation of China (Nos. U22B2049, 62302240), Fundamental Research Funds for the Central Universities, and Supercomputing Center of Nankai University (NKSC).

References

[1] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “Orb-slam: a versatile and accurate monocular slam system,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[2] J. L. Schonberger and J.-M. Frahm, “Structure-from-motion revisited,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4104–4113.

[3] G. Xiao, H. Wang, J. Ma, and D. Suter, “Segmentation by continuous latent semantic analysis for multi-structure model fitting,” *International Journal of Computer Vision*, vol. 129, no. 7, pp. 2034–2056, 2021.

[4] Z. Yang, Y. Yang, K. Yang, and Z.-Q. Wei, “Non-rigid image registration with dynamic gaussian component density and space curvature preservation,” *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2584–2598, 2018.

[5] G. Xiao, J. Ma, S. Wang, and C. Chen, “Deterministic model fitting by local-neighbor preservation and global-residual optimization,” *IEEE Transactions on Image Processing*, vol. 29, no. 4, pp. 8988–9001, 2020.

[6] C. Toft, W. Maddern, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, T. Pajdla *et al.*, “Long-term visual localization revisited,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 4, pp. 2074–2088, 2020.

[7] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[8] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *Proceedings of the European Conference on Computer Vision*, 2006, pp. 404–417.

[9] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: An efficient alternative to sift or surf,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 2564–2571.

[10] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, “Lift: Learned invariant feature transform,” in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 467–483.

[11] D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superpoint: Self-supervised interest point detection and description,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 224–236.

[12] J. Chang, J. Yu, and T. Zhang, “Structured epipolar matcher for local feature matching,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6176–6185.

[13] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, “Lightglue: Local feature matching at light speed,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 627–17 638.

[14] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[15] W.-Y. D. Lin, M.-M. Cheng, J. Lu, H. Yang, M. N. Do, and P. Torr, “Bilateral functions for global motion modeling,” in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 341–356.

[16] Y. Liu, L. Liu, C. Lin, Z. Dong, and W. Wang, “Learnable motion coherence for correspondence pruning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3237–3246.

[17] J. Bian, W.-Y. Lin, Y. Matsushita, S.-K. Yeung, T.-D. Nguyen, and M.-M. Cheng, “Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4181–4190.

[18] K. M. Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua, “Learning to find good correspondences,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2666–2674.

[19] G. Wang and Y. Chen, “Local consensus transformer for correspondence learning,” in *IEEE International Conference on Multimedia and Expo*, 2023, pp. 1151–1156.

[20] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[21] P. H. Torr and A. Zisserman, “Mlesac: A new robust estimator with application to estimating image geometry,” *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[22] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. John wiley & sons, 2005.

[23] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm, “Usac: A universal framework for random sample consensus,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 2022–2038, 2012.

[24] L. Zhou, S. Zhu, Z. Luo, T. Shen, R. Zhang, M. Zhen, T. Fang, and L. Quan, “Learning and matching multi-view descriptors for registration of point clouds,” in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 505–522.

[25] H.-Y. Chen, Y.-Y. Lin, and B.-Y. Chen, “Co-segmentation guided hough transform for robust feature matching,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 12, pp. 2388–2401, 2015.

[26] L. Cavalli, V. Larsson, M. R. Oswald, T. Sattler, and M. Pollefeys, “Handcrafted outlier detection revisited,” in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 770–787.

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

[28] L. Dai, Y. Liu, J. Ma, L. Wei, T. Lai, C. Yang, and R. Chen, “Ms2dg-net: Progressive correspondence learning via multiple sparse semantics dynamic graph,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8973–8982.

[29] C. Zhao, Z. Cao, C. Li, X. Li, and J. Yang, “Nm-net: Mining reliable neighbors for robust feature correspondences,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 215–224.

[30] W.-Y. Lin, F. Wang, M.-M. Cheng, S.-K. Yeung, P. H. Torr, M. N. Do, and J. Lu, “Code: Coherence based decision boundaries for feature correspondence,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 1, pp. 34–47, 2017.

[31] M. Cho and K. M. Lee, “Progressive graph matching: Making a move of graphs via probabilistic voting,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 398–405.

[32] C. Liu, S. Zhang, X. Yang, and J. Yan, “Self-supervised learning of visual graph matching,” in *Proceedings of the European Conference on Computer Vision*, 2022, pp. 370–388.

[33] P. C. Lusk, K. Fathian, and J. P. How, “Clipper: A graph-theoretic framework for robust data association,” in *IEEE International Conference on Robotics and Automation*, 2021, pp. 13 828–13 834.

[34] R. Wang, Z. Guo, S. Jiang, X. Yang, and J. Yan, “Deep learning of partial graph matching via differentiable top-k,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6272–6281.

[35] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv preprint arXiv:1609.02907*, 2016.

[36] X. Liu and J. Yang, “Progressive neighbor consistency mining for correspondence pruning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9527–9537.

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

[38] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 978–994, 2010.

[39] C. B. Choy, J. Gwak, S. Savarese, and M. Chandraker, "Universal correspondence network," *Advances in Neural Information Processing Systems*, pp. 2406–2414, 2016.

[40] I. Rocco, M. Cimpoi, R. Arandjelović, A. Torii, T. Pajdla, and J. Sivic, "Neighbourhood consensus networks," *Advances in Neural Information Processing Systems*, pp. 1658–1669, 2018.

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

[42] Q. Zhou, T. Sattler, and L. Leal-Taixe, "Patch2pix: Epipolar-guided pixel-level correspondences," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4669–4678.

[43] R. Wang, J. Yan, and X. Yang, "Combinatorial learning of robust deep graph matching: an embedding based approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 6984–7000, 2020.

[44] J. Lee, B. Kim, S. Kim, and M. Cho, "Learning rotation-equivariant features for visual correspondence," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 887–21 897.

[45] J. Ma, X. Jiang, A. Fan, J. Jiang, and J. Yan, "Image matching from handcrafted to deep features: A survey," *International Journal of Computer Vision*, vol. 129, no. 1, pp. 23–79, 2021.

[46] Y. Jin, D. Mishkin, A. Mishchuk, J. Matas, P. Fua, K. M. Yi, and E. Trulls, "Image matching across wide baselines: From paper to practice," *International Journal of Computer Vision*, vol. 129, no. 2, pp. 517–547, 2021.

[47] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.

[48] D. Barath, J. Matas, and J. Noskova, "Magsac: marginalizing sample consensus," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 197–10 205.

[49] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "Dsac-differentiable ransac for camera localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6684–6692.

[50] E. Brachmann and C. Rother, "Neural-guided ransac: Learning where to sample model hypotheses," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4322–4331.

[51] D. Barath, L. Cavalli, and M. Pollefeys, "Learning to find good models in ransac," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15 744–15 753.

[52] T. Wei, Y. Patel, A. Shekhovtsov, J. Matas, and D. Barath, "Generalized differentiable ransac," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 649–17 660.

[53] T. Wei, J. Matas, and D. Barath, "Adaptive reordering sampler with neurally guided magsac," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 18 163–18 173.

[54] C. Nie, G. Wang, Z. Liu, L. Cavalli, M. Pollefeys, and H. Wang, "Rlsac: Reinforcement learning enhanced sample consensus for end-to-end robust estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 9891–9900.

[55] C. Zhao, Z. Cao, J. Yang, K. Xian, and X. Li, "Image feature correspondence selection: a comparative study and a new contribution," *IEEE Transactions on Image Processing*, vol. 29, no. 2, pp. 3506–3519, 2020.

[56] X. Jiang, J. Ma, G. Xiao, Z. Shao, and X. Guo, "A review of multimodal image matching: Methods and applications," *Information Fusion*, vol. 73, pp. 22–71, 2021.

[57] R. Ranftl and V. Koltun, "Deep fundamental matrix estimation," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 284–299.

[58] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[59] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017, pp. 5099–5108.

[60] Z. Zhong, G. Xiao, L. Zheng, Y. Lu, and J. Ma, "T-net: Effective permutation-equivariant network for two-view correspondence learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1950–1959.

[61] S. Zhang and J. Ma, "Convmatch: Rethinking network design for two-view correspondence learning," in *AAAI Conference on Artificial Intelligence*, 2023, pp. 3472–3479.

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

[63] W. Sun, W. Jiang, E. Trulls, A. Tagliasacchi, and K. M. Yi, "Acne: Attentive context normalization for robust permutation-equivariant learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 286–11 295.

[64] X. Ye, W. Zhao, H. Lu, and Z. Cao, "Learning second-order attentive context for efficient correspondence pruning," in *AAAI Conference on Artificial Intelligence*, 2023, pp. 3250–3258.

[65] L. Zheng, G. Xiao, Z. Shi, S. Wang, and J. Ma, "Msa-net: Establishing reliable correspondences by multiscale attention network," *IEEE Transactions on Image Processing*, vol. 31, pp. 4598–4608, 2022.

[66] X. Liu, G. Xiao, R. Chen, and J. Ma, "Pgfnet: Preference-guided filtering network for two-view correspondence learning," *IEEE Transactions on Image Processing*, vol. 32, pp. 1367 – 1378, 2023.

[67] A. Myronenko, X. Song, and M. Carreira-Perpinan, "Non-rigid point set registration: Coherent point drift," *Advances in Neural Information Processing Systems*, pp. 1009–1016, 2006.

[68] Y. Liu, B. N. Zhao, S. Zhao, and L. Zhang, "Progressive motion coherence for remote sensing image matching," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022.

[69] J. Ma, J. Zhao, J. Jiang, H. Zhou, and X. Guo, "Locality preserving matching," *International Journal of Computer Vision*, vol. 127, no. 5, pp. 512–531, 2019.

[70] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International journal of computer vision*, vol. 60, pp. 63–86, 2004.

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

[72] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," in *Proceedings of the IEEE International Conference on Computer Vision*, 2005, pp. 1482–1489.

[73] J. You, J. Leskovec, K. He, and S. Xie, "Graph structure of neural networks," in *International Conference on Machine Learning*, 2020, pp. 10 881–10 891.

[74] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2020.

[75] L. Wu, Y. Chen, K. Shen, X. Guo, H. Gao, S. Li, J. Pei, B. Long *et al.*, "Graph neural networks for natural language processing: A survey," *Foundations and Trends® in Machine Learning*, vol. 16, no. 2, pp. 119–328, 2023.

[76] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *Acm Transactions on Graphics*, vol. 38, no. 5, pp. 1–12, 2019.

[77] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[78] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2net: A new multi-scale backbone architecture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 2, pp. 652–662, 2019.

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

[80] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, "Yfcc100m: The new data in multimedia research," *Communications of the ACM*, vol. 59, no. 2, pp. 64–73, 2016.

[81] J. Xiao, A. Owens, and A. Torralba, "Sun3d: A database of big spaces reconstructed using sfm and object labels," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1625–1632.

[82] O. Chum, T. Werner, and J. Matas, "Two-view geometry estimation unaffected by a dominant plane," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 772–779.

[83] D. Barath and J. Matas, "Graph-cut ransac," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6733–6741.

[84] D. Barath, J. Noskova, M. Ivashechkin, and J. Matas, "Magsac++, a fast, reliable and accurate robust estimator," in *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, 2020, pp. 1304–1312.

[85] J. Edstedt, I. Athanasiadis, M. Wadenbäck, and M. Felsberg, "Dkm: Dense kernelized feature matching for geometry estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2023, pp. 17 765–17 775.

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

[87] P. Truong, M. Danelljan, L. Van Gool, and R. Timofte, "Learning accurate dense correspondences and when to trust them," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5714–5724.

[88] H. Chen, Z. Luo, L. Zhou, Y. Tian, M. Zhen, T. Fang, D. Mckinnon, Y. Tsin, and L. Quan, "Aspanformer: Detector-free image matching with adaptive span transformer," in *Proceedings of the European Conference on Computer Vision*, 2022, pp. 20–36.

[89] D. Barath, D. Mishkin, M. Polic, W. Förstner, and J. Matas, "A large-scale homography benchmark," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 360–21 370.

[90] W.-L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C.-J. Hsieh, "Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 257–266.

[91] H. Li, M. Wang, S. Liu, P.-Y. Chen, and J. Xiong, "Generalization guarantee of training graph convolutional networks with graph topology sampling," in *International Conference on Machine Learning*, 2022, pp. 13 014–13 051.

[92] C. Wan, Y. Li, C. R. Wolfe, A. Kyrillidis, N. S. Kim, and Y. Lin, "PipeGCN: Efficient full-graph training of graph convolutional networks with pipelined feature communication," in *The International Conference on Learning Representations*, 2022, pp. 1–12.

[93] Y. Wu, J.-W. Liu, C.-Z. Zhu, Z.-F. Bai, Q.-G. Miao, W.-P. Ma, and M.-G. Gong, "Computational intelligence in remote sensing image registration: A survey," *International Journal of Automation and Computing*, vol. 18, pp. 1–17, 2021.

[94] B. Zitova and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, 2003.

[95] S. Chen, J. Chen, Y. Rao, X. Chen, X. Fan, H. Bai, L. Xing, C. Zhou, and Y. Yang, "A hierarchical consensus attention network for feature matching of remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2022.

[96] X. Jiang, J. Jiang, A. Fan, Z. Wang, and J. Ma, "Multiscale locality and rank preservation for robust feature matching of remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6462–6472, 2019.

[97] X. Jiang, J. Ma, A. Fan, H. Xu, G. Lin, T. Lu, and X. Tian, "Robust feature matching for remote sensing image registration via linear adaptive filtering," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 2, pp. 1577–1591, 2020.

[98] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3dmatch: Learning local geometric descriptors from rgb-d reconstructions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1802–1811.

[99] H. Wang, Y. Liu, Q. Hu, B. Wang, J. Chen, Z. Dong, Y. Guo, W. Wang, and B. Yang, "Roreg: Pairwise point cloud registration with oriented descriptors and local rotations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, pp. 10 376 – 10 393, 2023.

[100] X. Zhang, J. Yang, S. Zhang, and Y. Zhang, "3d registration with maximal cliques," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 745–17 754.

[101] Y. Wu, Y. Zhang, W. Ma, M. Gong, X. Fan, M. Zhang, A. Qin, and Q. Miao, "Rornet: Partial-to-partial registration network with reliable overlapping representations," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[102] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8958–8966.

[103] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *IEEE International Conference on Robotics and Automation*, 2009, pp. 3212–3217.

[104] X. Bai, Z. Luo, L. Zhou, H. Chen, L. Li, Z. Hu, H. Fu, and C.-L. Tai, "Pointdsc: Robust point cloud registration using deep spatial consistency," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 859–15 869.

[105] A. Schmied, T. Fischer, M. Danelljan, M. Pollefeys, and F. Yu, "R3d3: Dense 3d reconstruction of dynamic scenes from multiple cameras," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3216–3226.

[106] J. Zhang, D. Sun, Z. Luo, A. Yao, H. Chen, L. Zhou, T. Shen, Y. Chen, L. Quan, and H. Liao, "Oanet: Learning two-view correspondences and geometry using order-aware network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 3110–3122, 2022.

[107] M. Dusmanu, J. L. Schönberger, and M. Pollefeys, "Multi-view optimization of local feature geometry," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 670–686.

[108] V. Panek, Z. Kukelova, and T. Sattler, "Visual localization using imperfect 3d models from the internet," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 175–13 186.

[109] Z. Zhang, T. Sattler, and D. Scaramuzza, "Reference pose generation for long-term visual localization via learned features and view synthesis," *International Journal of Computer Vision*, vol. 129, pp. 821–844, 2021.

[110] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt, "Image retrieval for image-based localization revisited." in *British Machine Vision Conference*, vol. 1, no. 2, 2012, pp. 4–15.

[111] C. Toft, W. Maddern, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, T. Pajdla, F. Kahl, and T. Sattler, "Long-term visual localization revisited," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 4, pp. 2074–2088, 2022.

[112] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 716–12 725.

Xin Liu received the master's degree from the Department of Computer Science and Technology, Fujian Agriculture and Forestry University, Fuzhou, China, in 2022. He is currently pursuing the Ph.D. degree with Nankai University, Tianjin, China. His research interests include computer vision and image matching.

Rong Qin received the B.S.degree in the artificial intelligence from Chongqing University, Chongqing, China, in 2023. He is currently pursuing the Ph.D. degree with Nankai University, Tianjin, China. His current research interests include computer vision and high-resolution image processing.

Junchi Yan (S'10-M'11-SM'21) is the founding Deputy Director and Professor with School of Artificial Intelligence and Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. Before that, he was a Senior Research Staff Member with IBM Research where he started his career since April 2011. His research interests include machine learning and its applications. He regularly serves as Senior PC/Area Chair for NeurIPS, ICML, ICLR, CVPR, AAAI, IJCAI, SIGKDD, and Associate Editor for IEEE TPAMI, Pattern Recognition.

Jufeng Yang received the Ph.D. degree from Nankai University, Tianjin, China, in 2009. He is currently a Professor with the College of Computer Science, Nankai University and was a visiting scholar with the Vision and Learning Lab, University of California, Merced, USA, from 2015 to 2016. His recent interests include computer vision, machine learning and multimedia.