Knowledge-Based Systems 298 (2024) 111927

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

PMA-Net: Progressive multi-stage adaptive feature learning for two-view correspondence

Xiaojie Li [a,c,1], Fengyuan Zhuang [a,c,1], Yizhang Liu [d], Riqing Chen [a,b], Lifang Wei [a,b], Changcai Yang [a,b,*]

[a] Digital Fujian Research Institute of Big Data for Agriculture and Forestry, College of Computer and Information Sciences, Fujian Agriculture and Forestry University, Fuzhou 350002, China

[b] Sciences and Center for Agroforestry Mega Data Science, School of Future Technology, Fujian Agriculture and Forestry University, Fuzhou 350002, China

[c] Key Laboratory of Smart Agriculture and Forestry (Fujian Agriculture and Forestry University), Fujian Province University, Fuzhou 350002, China

[d] School of Software Engineering, Tongji University, Shanghai 201804, China

ARTICLE INFO

Keywords:
Feature matching
Outlier removal
Pose estimation
Local neighbor
Two-view correspondence

ABSTRACT

Establishing high-quality correspondences is a fundamental step for many computer vision tasks. Leveraging the local consistency of correct correspondences (i.e., inliers) has been a prevalent approach to distinguish them from incorrect correspondences (i.e., outliers). However, the random distribution of a significant proportion of outliers complicates the local neighborhood construction of inliers, making it inevitably contain many outliers, which compromises the reliability of the consistency information. In this paper, we propose a Progressive Multi-Stage Adaptive Feature Learning Network (PMA-Net) to solve this problem, which progressively and adaptively learns over multiple stages to remove random outliers embedded in the initial consistency. Specifically, we propose an Adaptive Dynamic Graph Construction (ADGC) module to construct a global topological graph with a high inlier ratio by calculating the affinity between the corresponding neighbors. Additionally, we also design a Feature Mapping Processing (FMP) block and Multi-Stage Prediction (MSP) block to improve the accuracy of subsequent local neighborhood information aggregation, as well as information loss due to channel dimension compression in the prediction phase. Experimental results in camera pose estimation, remote sensing image registration, and homography estimation demonstrate that the proposed PMA-Net performs better than other state-of-the-art methods on various public datasets. Code: https://github.com/XiaojieLi11/PMA-Net

1. Introduction

Two-view correspondence learning stands as a pivotal component in the field of computer vision, which can provide a foundation for downstream visual applications, such as 3D reconstruction [1], simultaneous location and mapping (SLAM) [2], image retrieval [3], image fusion [4], etc. The existing standard pipeline obtains the initial correspondences by detecting feature points and building the corresponding descriptors [5,6]. However, due to the ambiguity of feature descriptors, the strategy based on nearest-neighbor matching inevitably introduces incorrect correspondences into the initial correspondences. This challenge is notably pronounced, particularly in complex scenarios (e.g., viewpoint changes, illumination changes, occlusion, and image blurring). Therefore, many researchers have been dedicated to developing robust and accurate algorithms aimed at removing outliers while maintaining as many inliers as possible.

In general, inliers typically show consistency across image pairs, while outliers are randomly distributed [12]. Therefore, effective consistency has been widely studied for distinguishing inliers from outliers in recent years. Many traditional algorithms, such as [13–15], learn the potential local consistency of inliers by constructing the K-Nearest Neighbors (KNN) for each correspondence. However, traditional algorithms have some limitations when the matching scenarios are complex or have large-scale changes, especially when there are many outliers in initial correspondences. Fortunately, leveraging neural networks' powerful learning and representation capabilities, significant performance improvements have been achieved in addressing these

[*] Corresponding author at: Digital Fujian Research Institute of Big Data for Agriculture and Forestry, College of Computer and Information Sciences, Fujian Agriculture and Forestry University, Fuzhou 350002, China.

E-mail addresses: lxj15859819438@163.com (X. Li), zfy0505zsl@163.com (F. Zhuang), lyz8023lyp@gmail.com (Y. Liu), riqing.chen@fafu.edu.cn (R. Chen), weilifang1981@163.com (L. Wei), changcaiyang@gmail.com (C. Yang).

[1] Xiaojie Li and Fengyuan Zhuang contributed equally to this work.

https://doi.org/10.1016/j.knosys.2024.111927
Received 1 November 2023; Received in revised form 2 April 2024; Accepted 8 May 2024
Available online 18 May 2024

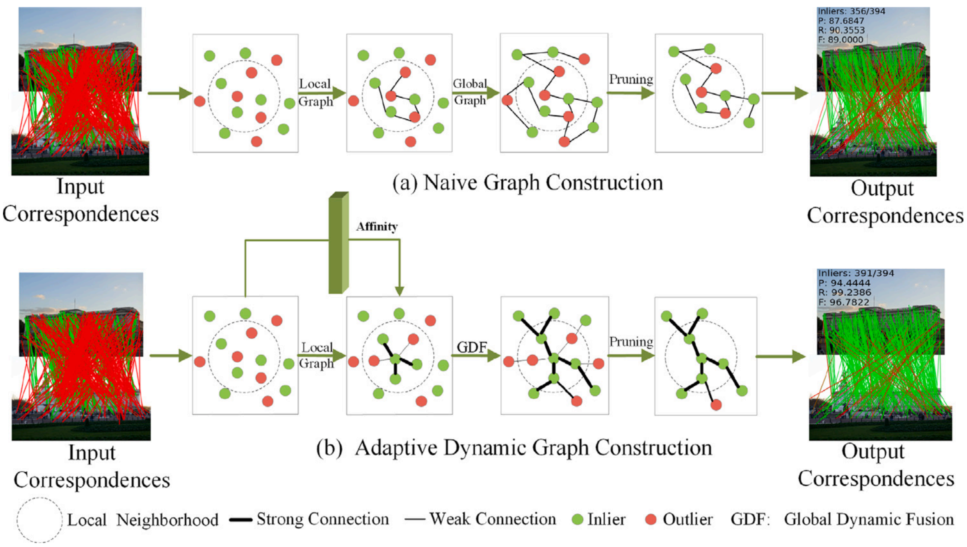

Fig. 1. The composition of different correspondence learning methods. The points shown in the figure represent keypoints extracted from given initial correspondences, the green/red point indicates the correct/incorrect correspondence. (a) [7–11] learn the potential local consistency of inliers by constructing KNN for each correspondence. (b) Our PMA-Net calculates the affinity between correspondence and adjacency relationships during the graph construction to discover different types of neighbors (inliers/outliers).

challenges. NM-Net [7] proposes a compatibility-specific local graph context mining method that hierarchically extracts and aggregates local correspondences to find consistent K neighbors. MS^{2}DG-Net [8] designs sparse semantic dynamic graphs to capture local topological information between correspondences. CLNet [10] uses KNN and GCN to build network graph to mine local-to-global neighborhood relationships to search and aggregate local context. NCMNet [11] explores graph context information in coordinates, features, and global neighborhoods to guide corresponding pruning. All of the above methods capture the consistency between inliers by constructing local neighborhood graphs, as shown in Fig. 1(a). Although the above networks perform well, there are still some outliers that appear around inliers. The explicit neighbor relation modeling based on the Euclidean distance metric effectively removes the outliers that are free from the outside (Fig. 1(a)). However, the outliers that are included in K-nearest neighbors during graph construction cannot be removed, resulting in substantial neighbor interference during subsequent local neighborhood graph construction. If we can distinguish the connection between the inliers and inliers, the inliers and outliers, and the outliers and outliers in the composition, this problem may be solved well. Inspired by the Graph Attention Network [16], we design an Adaptive Dynamic Graph Construction (ADGC) module to calculate the affinity between nodes and their neighbors, as shown in Fig. 1(b). The affinity between inlier and inlier is more significant than the affinity between inlier and outlier. Then, we can use these affinities to remove outliers and build a clean global topological graph.

Besides aforementioned issues, another crucial consideration is how to ensure that the feature correspondences remain uncontaminated before local neighborhood graph construction? Many learning-based methods [8,10,17–21] directly map the initial correspondences to high-dimensional space in channel, resulting in a large amount of redundant feature being introduced. In this paper, we design a Feature Mapping Processing (FMP) block to preprocess the initial features mapped to the high-dimensional space, thereby alleviating the interference to the subsequent construction of local neighborhood graph. In addition, during the prediction stage, direct compression of channel dimension will result in the loss of important details and structural information [8,10,17,18,20,22,23]. We propose the Multi Stage Prediction (MSP) block to gradually separate geometric topological features and disentangle the intertwined information to predict the probability of inliers.

Our contributions are as follows:

- Firstly, we propose an ADGC module to calculate the affinity between correspondence and its neighbors by considering the dynamic changes and correlations of features, thus constructing the global topological graph to remove outliers.
- Secondly, we propose a FMP block and an MSP block to improve the accuracy of subsequent local neighborhood information aggregation, and information loss caused by the compression of channel dimension in the prediction stage, respectively.
- Finally, we design a novel Progressive Multi-Stage Adaptive Feature Learning Network (PMA-Net) to effectively remove outliers embedded in the inlier consistency. Experimental results on various public datasets demonstrate that our proposed PMA-Net performs better than the current advanced image matching methods.

2. Related work

2.1. Traditional methods

RANSAC (Random Sample Consensus) [24], a classic and traditional correspondence extraction method based on local consistency, aims to estimate model parameters and exclude outliers by iteratively finding the best feature fit through random sampling and stepwise approximation. In order to improve efficiency and accuracy, many improved methods have emerged based on this. For example, DEGENSAC [25] can identify and screen reliable models more accurately by introducing H-degradability algorithms. USAC [26] adopts an adaptive approach in sampling strategy, model evaluation and threshold selection, which can provide higher performance and better robustness. MLESAC [27] combines the idea of maximum likelihood estimation based on RANSAC, improves the robustness of parameter estimation and outlier rejection by calculating the likelihood of the feature on the basis of the hypothetical model. However, due to the sensitivity of the RANSAC to parameter settings and its dependence on feature sampling, it may generate unstable results and long computation time.

2.2. Learning-based methods

Unlike traditional correspondence methods, the learning-based feature matching method learns how to select reliable correspondence and eliminate incorrect correspondence by putting the initial correspondence set into a neural network for training, which is more competitive in the face of more complex features with low inlier rates. As a pioneering exploration, LFGC [17] uses a multilayer perceptron to weight each pair of candidate matching points. OANet [18] utilizes sparse mapping to make sequential awareness of complex global contexts in a learnable manner. Superglue [28] utilizes a graph neural network with attention for local feature matching, showcasing powerful functionality. VLSG-SANet [29] extracts the local structural information of each feature point by CNN to generate visual descriptors to capture the multi-scale topology of each feature point. SemLA [30] embeds semantic information explicitly at various stages of the network in a coordinated manner and achieves good performance through the design of effective semantic object feature matching paradigms and fusion methods. JRA-Net [31] introduces the utilization of joint representations at different scales to mitigate trivial correspondences, thereby establishing robust correspondences in image pairs. ConvMatch [32], leverages Convolutional Neural Networks (CNNs) to construct a motion field for correcting errors induced by local outliers. PGFNet [20], in its approach, devises a novel iterative filtering structure to learn preference scores for correspondences, thereby guiding subsequent matching operations. These methodologies effectively select reliable correspondences and eliminate mismatched pairs by employing well-designed deep neural networks. However, it is worth noting that they do not explicitly take into account the extrinsic geometry and isomorphic relationships between keypoints, leaving them susceptible to the influence of outliers during the network learning process.

2.3. Consistency of correspondences

In the context of extrinsic geometry or isomorphism constraints, it is paramount to emphasize that valid correspondences exhibit mutual agreement, as they conform rigorously to specific geometric principles and isomorphic relationships. Conversely, incorrect correspondences lack such alignment due to their stochastic nature, as they deviate from any well-defined geometric criteria. Consequently, the principle of consistency has guided the development of several commendable algorithms, such as LPM [13], which utilizes distance and topological constraints to ensure the relative positional consistency of correspondences before and after transformation, thereby optimizing the matching results. In order to extract and aggregate more reliable features, NM-Net [7] adopts the compatibility-specific K-nearest neighbor mining method to search for local consistent neighbors. GCSNMatcher [33] uses geometric information between sparse points to construct local neighborhood graph structures. CLNet [10] introduces a correspondence pruning algorithm transitioning from local KNN to global GCN, acquiring high-quality matching sets. LMCNet [9] presents local motion consistency by fusing KNN maps with max pooling and employs a smoothing function to fit the global motion consistency. MS^{2}DGNet [8], alternatively, leverages K-nearest neighbor algorithms to construct dynamic sparse semantic information graphs for domain information acquisition. MNCNet [11] constructs coordinates, space, and global neighbors to mine for consistency between counterparts. However, these algorithms treat all neighbor nodes equally when constructing graphs using KNN, introducing uncertainty in selecting the optimal neighboring nodes. We propose an Adaptive Dynamic Graph Construction (ADGC) Module to construct a global topological graph with a high ratio inliers by calculating the affinity between the corresponding neighbors.

3. Proposed method

In this section, we will introduce the components of PMA-Net, including the problem formulation, the FMP block, the ADGC module, the MSP block, and the Loss function. The pipeline is shown in Fig. 2.

3.1. Problem formulation

Given a pair of images (I, I'), we use typical feature extraction methods (e.g., SIFT [5] or SuperPoint [6]) to detect keypoints and construct corresponding descriptors. On this basis, an initial correspondence set $S = \left\{s_1, s_2, \dots, s_N\right\} \in \mathbb{R}^{N\times 4}$ is established by using the nearest neighbor matching strategy of feature descriptors, where n represents the number of correspondences and $s_i = (x_i, y_i, x_i{}', y_i{}')$ is the ith correspondence between two keypoints (x_i, y_i) and $(x_i{}', y_i{}')$ in I and I', respectively, both of which are normalized with the camera intrinsic. Due to the ambiguity of feature descriptors, strategies based on nearest neighbor matching will inevitably introduce many incorrect correspondences in the initial correspondences. This challenge is particularly evident, especially in complex scenarios involving changes in viewpoint, lighting, occlusion, and image blurring, such as the YFCC100M [34] and SUN3D datasets [35]. Therefore, our goal is to remove incorrect correspondences and recover the camera pose accurately.

To this end, we develop an effective Progressive Multi-stage Adaptive Feature Learning network (PMA-Net), as shown in Fig. 1. Following previous works [8,17,18], we use an iterative learning strategy to train PMA-Net to produce the fine probability set P'. Firstly, initial correspondence set S is fed into the first iterative network to calculate coarse inlier probability set P_j and residual set R_j. Subsequently, we put the initial correspondence set S, the residual set R_j and the coarse inlier probability set P_j into next iteration network to compute fine inlier probability, where R_j and P_j are the guiding information. Finally, we can obtain a fine inlier probability set $P' = [p'_1, p'_2, \dots, p'_N] \in \mathbb{R}^{N\times 1}, p'_i \in [0, 1)$. The above framework can be expressed as:

$$P_j = \mathcal{F}_1(S), \tag{1}$$

$$R_j = ep(S, g(S, P_j)), \tag{2}$$

$$P' = \mathcal{F}_j\left(\left[P_{j-1} \middle\| R_{j-1} \middle\| S\right]\right), j = 2, 3, \dots, n, \tag{3}$$

where $\mathcal{F}_1(\cdot)$ and $\mathcal{F}_j(\cdot)$ represents the first iteration network and the jth iteration network, respectively. n is the number of iterations. $ep(\cdot)$ is the epipolar error calculation operation. $[\cdot \| \cdot]$ represents the concatenate operation in the channel dimension. Then, the fine probability set P' is put into the weighted eight-point algorithm to calculate the essential matrix E' [17]. The operation can be written as:

$$E' = g(S, P'), \tag{4}$$

where $g(\cdot)$ is the weighted eight-point algorithm.

3.2. Feature Mapping Processing (FMP) block

To enhance the accuracy of subsequent local neighborhood information aggregation, we design the Feature Map Processing (FMP) block aimed at preprocessing the initial features mapped to higher-dimensional space, as shown in Fig. 3. The proposed FMP block comprises two branches, the first branch uses spatial attention [36] to learn the matched spatial feature consistency F^s, which explore the weights of different features for the subsequent construction of local neighborhood graphs. Meanwhile, the second branch uses the scale factor γ, embedded in Batch Normalization [37], to suppress the redundant feature generated by channels being mapped to high dimensions. Specifically, the scale factor γ, post-normalization, configures a weight matrix W_γ that is then multiplied by the features S^γ after Batch Normalization to calculate the channel weights F^c. This method is adaptive in adjusting the weights of different channel features, reducing feature redundancy while maintaining stable performance. In contrast to self-attention mechanisms, the second branch does not escalate network complexity and requires a surplus of additional parameters while implementing feature weighting. Ultimately, through the interplay of these

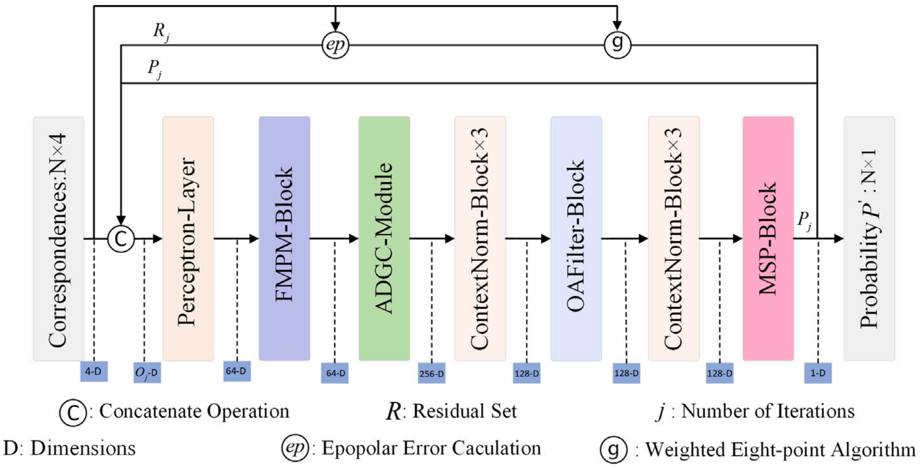

Fig. 2. Pipeline of the proposed PMA-Net. The dimensions of both R and P are $N \times 1$. O_j represents the channel dimension of this position in the ith iteration, when $j = 1, M_j = 4$; $j \geq 2$, $O_j = 6$.

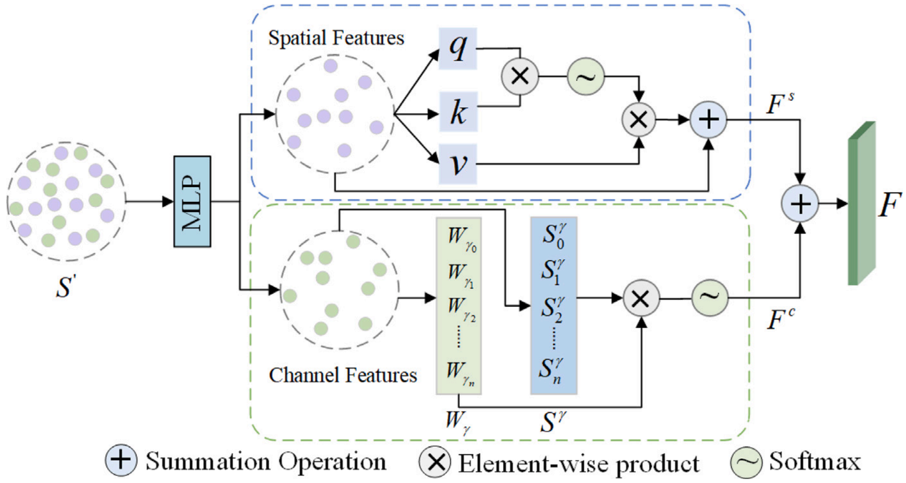

Fig. 3. The illustration of the Feature Mapping Processing (FMP) Block.

two branches, we obtain more refined feature representations F of the correspondences in spatial and channels. The above operation can be written as:

$$S' = MLP(S), \tag{5}$$

$$F^s = SA(S'), \tag{6}$$

$$S^\gamma = \gamma \frac{S' - \mu_{S'}}{\sqrt{\sigma_{S'}^2 + \varepsilon}} + \beta, \tag{7}$$

$$W_{\gamma_i} = \frac{\gamma_i}{\sum_{i=0}^{N} \gamma_i}, \tag{8}$$

$$F^c = sigmoid(W_\gamma \times S^\gamma), \tag{9}$$

$$F = F^s + F^c, \tag{10}$$

where $MLP(\cdot)$ denotes an MLP layer. S' represents features mapped into high dimensions. $SA(\cdot)$ and $CA(\cdot)$ represent spatial and channel attention, respectively. The scale factor γ and β are learnable parameters. $\mu_{S'}$ and $\sigma_{S'}^2$ are the mean and the variance, respectively. W_{γ_i} is the weight matrix.

3.3. Adaptive Dynamic Graph Construction (ADGC) module

To accurately construct a global topological graph, we design a novel Adaptive Dynamic Graph Construction (ADGC) module that adaptively selects and adjusts graph construction by calculating the affinity between nodes and their neighbors. It consists of three components: Local Neighborhood Graph Construction, Neighbor Selection, and Global Dynamic Fusion. The framework is illustrated in Fig. 4.

3.3.1. Local neighborhood graph construction

We firstly search the K-nearest neighbors for $f_i \in F$ and obtain a set of nodes $V_i = \{f_i^k, k = 1, 2, \ldots, K\}$. Then, we link f_i and its K-nearest neighbors $\{f_i^k, k = 1, 2, \ldots, K\}$ to construct directed edge set E_i^k. Following [8,10], we describe the directed edge E_i^k as

$$E_i^k = \left[f_i \middle\| f_i - f_i^k\right], \tag{11}$$

where $f_i - f_i^k$ is the residual feature, $[\cdot\|\cdot]$ represents the concatenation operation along the channel dimension. We can construct a local neighborhood graph $G_i = (V_i, E_i^k) \in \mathbb{R}^{N \times C \times K}$ for each feature f_i.

3.3.2. Neighbor selection

After constructing the local neighborhood graph, we design a simple yet efficient method to calculate the affinity between each node and its neighbors, reflecting the interrelatedness of each node with its

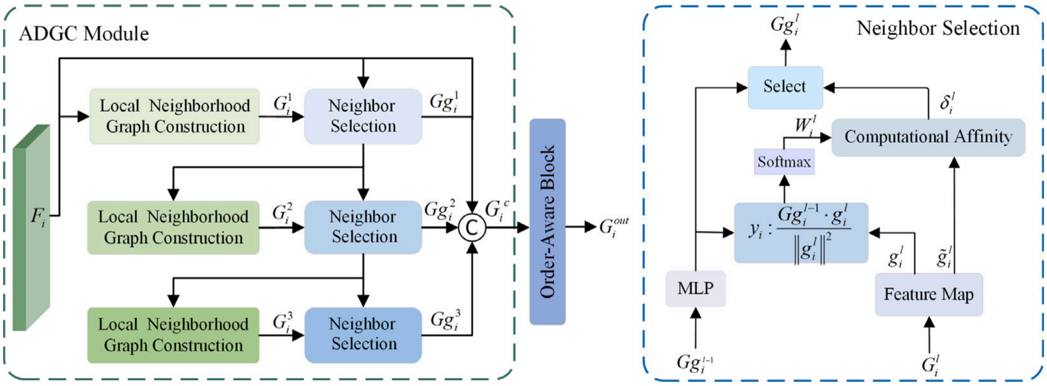

Fig. 4. The Architecture of Adaptive Dynamic Graph Construction (ADGC) module. Gg_i^0 indicates the initial feature F_i. In addition, the Order-Aware Block consists of the ContextNorm [17] block and OAFilter [18] block.

neighboring nodes, as shown on the Neighbor Selection block in Fig. 4. Specifically, we can obtain the local neighborhood graph $G_i^l \in \mathbb{R}^{N\times C\times K}$ from the lth Local Neighborhood Graph Construction on the ADGC module. We input it to the Neighbor Selection and utilize the PointCN layer [38] to obtain the graph vectors g_i^l and $\widetilde{g}_i^{\,l}$. Meanwhile, the local neighborhood topological graph Gg_i^{l-1} where $Gg_i^0 = F_i$ is used as the projection vector of the lth Neighbor Selection. The projection y_i of Gg_i^{l-1} can be calculated by using the graph vectors g_i^l. Then, the weight $W_i^l \in \mathbb{R}^{N\times C\times K}$ can be obtained by using softmax function to recalibrate the projection y_i. The above operations can be written as:

$$(g_i^l, \widetilde{g}_i^{\,l}) = PointCN(G_i^l), \tag{12}$$

$$y_i = \frac{Gg_i^{l-1} \cdot g_i^l}{\left\| g_i^l \right\|^2}, \tag{13}$$

$$W_i^l = Softmax(y_i), \tag{14}$$

where $PointCN(\cdot)$ denotes PointCN layer [38], $\|\cdot\|$ denotes L2-norm. Furthermore, the weight $W_i^l \in \mathbb{R}^{N\times C\times K}$ are element-wise multiplied with $\widetilde{g}_i^{\,l}$ to calculate local topological graph affinity $\delta_i{}^l$.

$$\delta_i^l = W_i^l \odot \widetilde{g}_i^{\,l}, \tag{15}$$

where $\odot$ denotes Hadamard product. Subsequently, based on the affinity, we adaptively select a subset of nodes with the highest affinity, which is used to construct the local neighborhood topological graph. Finally, we use a ResNet [39] module to enhance the feature expression capability. Based on the above steps, we can get the local neighborhood graph Gg_i^l with affinity between nodes.

$$Gg_i^l = ResNet(Sel(\delta_i^l, G_i^l)), \tag{16}$$

where $Sel(\cdot)$ selects a subset of nodes with the highest affinity.

3.3.3. Global dynamic fusion

However, Gg_i^l only contains the local neighborhood information. One method for obtaining global graph information is by expanding the receptive field. The receptive field size of the topological graph is contingent upon the neighborhood K. Arbitrarily inflating the neighborhood K will result in a considerable escalation of computational resources which is impractical for real-world applications. Inspired by multi-layer convolutional neural networks [40,41], which divide large kernels into smaller ones in a cascaded manner to capture overall features. Therefore, we implement the method of cascade local neighborhood topological graph to obtain the global graph information. The process of each dynamic iteration enables the model to adaptively learn the global affinity between neighboring nodes at each stage, thereby allowing for a more comprehensive fusion of global topological graph information. We assume that there are L Local Neighborhood Graph Construction operations and Neighbor Selection operations cascaded, gradually reinforcing the learning of neighbor through multi-level iterations. Subsequently, Gg_i^l is concatenated with the previously established local neighborhood graph, Gg_i^{l-1}, along the channel dimension. This process facilitates the fusion of local neighborhood topological graphs, culminating in the formation of a coarse global topological graph G_i^c.

$$G_i^c = \left[F_i \left\| Gg_i^1 \right\| \dots \left\| Gg_i^L \right. \right], \tag{17}$$

where L represents the number of cascades. Ultimately, we employ ContextNorm blocks [17] in conjunction with OAFilter blocks [18] to fuse topological information from various stages, thereby generating the global topological graph G_i^{out}.

3.4. Multi-Stage Prediction (MSP) block

Once the global topological graph G^{out} are constructed, we need to consider how to effectively predict inliers probabilities. Different from other methods (e.g., [8,10,17,18,20,22,23]), we do not directly compress features from 128 dimensions to 1 dimension in the prediction probability stage, which result in the loss of crucial details and structural information, thereby reducing feature expressiveness. We design a Multi-Stage Prediction block with packet compression to gradually separate global topological features. The MSP block comprises two ResNet blocks and three convolution layers with 1 × 1 kernels, as illustrated in Fig. 5. By leveraging the ResNet block [39] to enhance feature representation and alleviate gradient vanishing, we further reduce information loss through small-batch grouping compression, subsequently separating G^{out} into two distinct components z_c and z_k along the channel dimension. Immediately following instance normalization and batch normalization, we apply a convolutional operation to compress the data from D-dimensional channels to a more compact $(D/2)$-dimensions. After merging the two components, we obtain a more compact feature representation G'. Then, we repeat the above process to compress the channel once again to $(D/4)$-dimensions. Finally, we use a convolution to reduce the channel dimension to 1. The operation can be written as:

$$(z_c, z_k) = Spl(ResNet(G^{out})), \tag{18}$$

$$G' = [\vartheta(z_c) \,\|\, \vartheta(z_k)], \tag{19}$$

$$(z_c{}', z_k{}') = Spl(ResNet(G')), \tag{20}$$

$$G'' = [\vartheta(z_c') \,\|\, \vartheta(z_k')], \tag{21}$$

$$P_{msp} = Conv(G''), \tag{22}$$

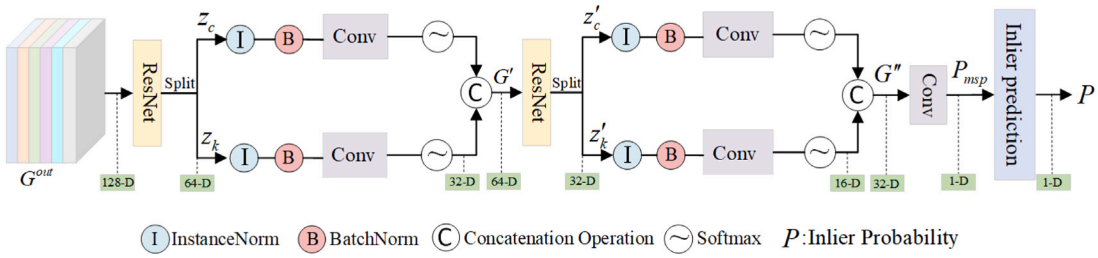

Fig. 5. The illustration of the Multi-Stage Prediction Block.

where G^{out} is the global topological graph obtained by the ADGC module. $Spl(\cdot)$ is a split operation. $\vartheta(\cdot)$ represents the compression operation, including Instance Normalization, Batch Normalization, Convolution, and Softmax operations. $z_c{}'$ and $z_k{}'$ are two distinct components obtained by splitting G' along the channel dimension. $[\cdot \| \cdot]$ represents the concatenation operation along the channel dimension. G' and G'' are the features obtained by the first compression and the second compression, respectively. $Conv(\cdot)$ is convolution with 1×1 kernels. According to P_{msp}, we can predict the inlier probability and calculate the rotation and translation matrix by the weighted eight-point algorithm.

3.5. Loss function

We utilize a hybrid loss function to optimize the proposed network:

$$Loss = L_p(P', T) + \tau L_e(E, E'), \tag{23}$$

where $L_p(\cdot,\cdot)$ represents the binary cross-entropy loss function used for classification operations. P' and T represent the predicted probability set and weakly supervised ground truth (labels). $L_e(\cdot,\cdot)$ represents the error calculation of the regression essential matrix, where E is the ground truth essential matrix, E' is the essential matrix for prediction. τ is the weight parameter to balance the two losses.

3.6. Implementation details

In our implementation, PMA-Net performs two iterations and the number of m in the ADGC module is 4, as shown in Fig. 2. After the first iteration, we use probability P_1 as a pruning guide weight to eliminate outliers in the initial correspondence set S to alleviate the influence of outliers during the next iteration. We choose two learning-based networks (OANet++ [18] and MS^{2}DGNet [8]) as baselines. In addition, we employ an Adam optimizer [42] with an appropriate learning rate strategy, adopting a progressive strategy during training. Specifically, we gradually increase the learning rate with a linear increment for the first 10,000 iterations. Subsequently, we reduce the learning rate by a factor of 0.4 every 20,000 iterations, with a total of 50,000 iterations. The batch size is set to 32 for each iteration. Our experimental platform runs on the Ubuntu 18.04 operating system and utilizes an NVIDIA GTX 3090.

4. Experiment results

To validate the effectiveness of PMA-Net in various challenging tasks and scenarios, we conduct a series of comparative experiments for outlier removal, camera poses estimation, remote sensing image registration, and homography estimation. In addition, we conduct ablation studies to verify the effectiveness of each component in our proposed PMA-Net.

4.1. Dataset

4.1.1. Outdoor dataset

We utilize the YFCC100M dataset released by the Yahoo research team [34] which includes 100 million images from the internet as the outdoor dataset. The dataset is divided into 72 sequences based on different scene content. We select 68 sequences as the training set and the rest as the test set.

4.1.2. Indoor dataset

We use the SUN3D dataset [35] which is mainly used for scene understanding and 3D reconstruction as the indoor dataset. It includes RGB-D image sequences of indoor environments, consisting of 254 indoor image sequences with sparse textures, overlapping elements. The indoor dataset has 239 sequences for training and validation, and 15 sequences for testing.

For both datasets mentioned above, the test set is considered as an unknown scene, while 20% of the training set is considered as a known scene for evaluating outlier removal and camera pose estimation.

4.1.3. Remote sensing image dataset

We select 60 pairs of images from different scenes in the remote sensing image dataset [43] for the registration experiment. The dataset consists of Synthetic Aperture Radar (SAR), Color Infrared Aviation (CIAP), Unmanned Aerial Vehicle (UAV), and 720Yun datasets, which have different image transformations such as non-rigid transformations, terrain undulations, occlusion and noise, and resolution differences, making matching difficult.

4.1.4. HPatches dataset

The HPatches dataset [44] contains a total of 696 images, including 116 different scenes with six images for each scene. Fifty-seven scenes are taken under different lighting conditions, while other scenes with viewpoint changes have issues such as rotation, scale change, brightness change, blur, and noise. This dataset is used for homology matrix estimation.

4.2. Camera pose estimation

Camera pose estimation is an essential task in computer vision, aiming to recover camera pose in three-dimensional space by analyzing feature points within the image, i.e., camera rotation and translation information. For our experiments, we extract up to 2000 input putative correspondences using SIFT or SuperPoint operators for each image in the YFCC100M dataset [34] and the SUN3D dataset [45]. These correspondences are then put into our proposed PMA-Net for matching.

To evaluate the performance of PMA-Net, we adopt the mean average precision (mAP) of the angular differences between the rotation/translation predicted by the network and the ground truth as the error metric and choose mAP under 5° as the default metric. We compare PMA-Net with several advanced traditional and learning-based methods, such as RANSAC [24], Point-Net++ [38], ACNe [35], LMCNet [17], OA-Net+++ [46], SuperGlue [28], CLNet [10], T-Net [47],

Table 1
Camera pose estimation results using the SIFT operator on the YFCC100M dataset and SUN3D dataset. The mAP5° (%) without/with RANSAC are reported. The bold ones are the best.

mAP5°	Size (MB)	YFCC100M (%)		SUN3D (%)	
		Known	Unknown	Known	Unknown
RANSAC [24]	–	–/5.82	–/9.08	–/4.38	–/2.86
Point-Net++ [38]	12.00	10.49/33.78	16.48/46.25	10.58/19.17	8.10/15.29
ACNe [35]	0.41	25.55/39.08	35.40/51.62	13.44/21.08	11.62/16.40
LMCNet [17]	0.93	33.73/40.39	47.50/55.03	19.92/21.79	16.82/17.38
OANet++ [18]	2.47	31.00/41.40	35.07/51.45	19.22/22.29	13.69/22.29
SuperGlue [28]	12.02	35.00/43.17	48.12/55.06	22.50/23.68	17.11/18.23
CLNet [10]	1.26	39.00/45.22	54.05/59.70	21.34/23.85	17.14/18.13
T-Net [47]	3.78	42.99/45.25	48.20/55.85	22.38/22.96	17.24/17.57
MSA-Net [48]	1.45	39.53/44.57	50.65/56.28	18.64/22.03	16.86/17.79
PGFNet [20]	2.99	44.20/46.28	53.70/57.83	23.66/23.87	19.32/18.00
MS^{2}DGNet [8]	2.61	38.36/45.34	49.13/57.68	22.20/23.00	17.84/17.79
PMA-Net(ours)	2.96	**49.42/49.30**	**61.48/62.00**	**29.02/23.95**	**21.10/18.80**

MSA-Net [48], PGFNet [20], and MS^{2}DGNet [8]. Table 1 gives the camera pose estimation results using the SIFT operator on the YFCC100M dataset and the SUN3D dataset. From Table 1, we can see that PMA-Net achieves optimal performance. In particular, compared with the baseline network MS^{2}DGNet, PMA-Net increases mAP5° by 11.06% and 12.35% for known scene and unknown scene in the YFCC100M dataset without RANSAC, respectively. Additionally, we observe that using RANSAC as a subsequent processing step may lead to a performance decrease. For the SUN3D dataset, the performance of PMA-Net with RANSAC decreases by 5.07% and 2.30% compared to the results without RANSAC. This is due to the highly repetitive nature, significant scale variations, and the existence of textureless regions in indoor scenes (SUN3D dataset), resulting in the presence of numerous outliers. The operational principle of RANSAC relies on the generation and testing of estimates from random subsets, and in situations where there is a substantial number of outliers in the initial correspondences, it may introduce more outliers, causing convergence to models of lower accuracy and consequently resulting in poorer performance than without RANSAC.

Table 2 gives the camera pose estimation results using the SuperPoint descriptor on the YFCC100M and SUN3D dataset. For the SuperPoint descriptor, the results provided by PMA-Net are better than other methods. The PMA-Net improves by 8.64% and 12.1% compared with the MS^{2}DGNet for known scenes and unknown scenes in the YFCC100M dataset without RANSAC, respectively. Moreover, the camera pose estimation results obtained by the SuperPoint descriptor are worse than those obtained by SIFT due to the lower accuracy of SuperPoint's keypoint localization, which hinders the recovery of accurate camera poses. From the comparative results of network performance and computational complexity in Tables 1 and 2, we can note that our PMA-Net achieves the best performance within reasonable computational complexity.

In addition, Fig. 6 shows partial typical visualization matching results of RANSAC, MS^{2}DGNet, and our PMA-Net. From Fig. 6, we can see that the matching results of PMA-Net are better than those obtained by RANSAC and MS^{2}DGNet.

4.3. Remote sensing image registration

The key step of image registration is establishing high-quality correspondences to locate the overlapping region between two images. Here, we perform registration experiments to evaluate the performance of PMA-Net. We use the networks trained on the YFCC100M dataset to remove incorrect correspondences from the matching set and ensure more accurate and reliable registration effects. Following [43,49], Root Mean Square Error (RMSE), Mean Euclidean Error (MEE), Mean Absolute Error (MAE), and Running Time (RT) are used as evaluation metrics to assess the performance of the registration methods. We compare our PMA-Net with other methods including RANSAC [24], LPM [13], OANet++ [18], CLNet [10], and MS^{2}DGNet [8]. RANSAC and LPM are traditional methods, and the rest are learning-based methods. The registration results obtained by different methods are presented in Table 3 and Fig. 7. From Table 3 and Fig. 8, we can see that the performance of RANSAC is not satisfactory because RANSAC requires a high ratio of inliers and is not suitable for remote sensing image registration with high outliers. OANet++ and CLNet take less time, but their performance is not good compared to other learning-based algorithms. PMA-Net performs best in terms of RMSE, MAE, and MEE. Fig. 8 shows the remote sensing image registration results by different methods. We can find that the results from RANSAC are distorted and deformed to varying degrees in these four scenes. Other methods also exhibit distortion in different scenarios, such as LPM in the 720Yun image and MS^{2}DGNet in the CIAP image. The results by our proposed algorithm (the last row) maintain good visualization results in all scenarios.

4.4. Homography estimation

Homography estimation is one of the basic experiments of feature matching. The geometric transformation relationship between two images is estimated by calculating the homography matrix. We use the public HPatches dataset [44], which provides a series of image pairs of natural scenes with different scenes and angle variations, and each pair of images has a corresponding homography matrix for reference. Specifically, the SIFT is used to extract 4000 keypoints from the HPatches and establish the initial correspondences. Then, the outlier removal methods, such as our PMA-Net, are used to predict inliers' probability from initial correspondences. The inliers will be obtained according to the predicted inliers' probability and used to calculate the homography matrix. We compare our PMA-Net with other methods including PointCN [17], OANet++ [18], CLNet [10], LMCNet [9], and MS^{2}DGNet [8]. Following [6], the percentage of correct homography estimation in which the reprojection error is less than 3/5/10 pixels ($ACC.@3/5/10PX$) is used to compare the performance of homography estimation. Table 4 gives the $ACC.@3/5/10PX$ by different methods. As illustrated in Table 4, our PMA-Net achieves the best results for all thresholds. This is due to PMA-Net can progressively and comprehensively learn local neighbor information across multiple stages, effectively eliminating outliers, and ultimately minimizing the average position deviation of image feature points calculated by the homography matrix.

4.5. Ablation studies

4.5.1. Main components

To demonstrate the effectiveness of each component of the proposed PMA-Net, we conduct an ablation study on the FMP block, ADGC module, and MSP block. MS^{2}DGNet is the baseline. Table 5 shows

Table 2
Camera pose estimation results using the SuperPoint descriptor on the YFCC100M dataset and SUN3D dataset. The mAP5° (%) without/with RANSAC are reported. The bold ones are the best.

mAP5°	Size (MB)	YFCC100M (%)		SUN3D (%)	
		Known	Unknown	Known	Unknown
RANSAC [24]	–	–/12.85	–/17.47	–/14.93	–/12.15
Point-Net++ [38]	12.00	11.87/28.46	17.95/38.83	11.40/21.19	9.38/17.08
ACNe [35]	0.41	26.72/31.16	32.98/45.34	18.35/21.12	13.82/18.05
LFGC [17]	0.39	12.18/30.25	24.25/42.57	12.63/21.81	10.68/17.36
OANet++ [18]	2.47	29.52/35.72	35.27/45.45	20.01/24.43	15.62/18.56
CLNet [10]	1.26	27.93/32.75	38.48/45.02	16.15/23.56	13.62/18.58
T-Net [47]	3.78	35.73/37.99	40.62/46.375	21.62/24.66	17.18/19.09
MSA-Net [48]	1.45	30.63/-	38.53/47.50	–	–
PGFNet [20]	2.99	33.96/37.09	42.03/47.30	19.78/24.21	15.39/18.82
MS^{2}DGNet [8]	2.61	30.40/36.02	37.38/46.48	20.28/24.86	16.08/18.67
PMA-Net(ours)	2.96	**39.07/39.85**	**49.48/52.18**	**24.05/25.74**	**18.24/19.18**

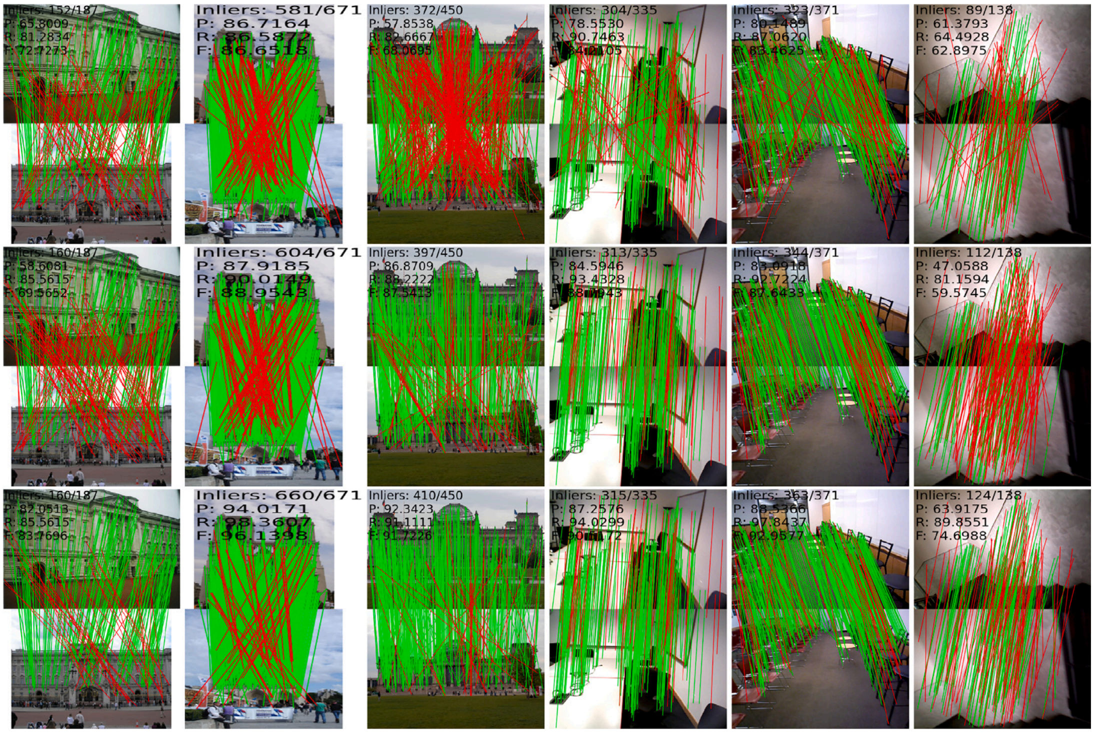

Fig. 6. Partial typical visualization results of RANSAC, MS^{2}DGNet and PMA-Net (from top to bottom). The first three columns are from the YFCC100M dataset. The remaining images are from the SUN3D dataset. The green lines and red lines represent inliers and outliers, respectively. The inlier rate, P, R, and F values of each picture are identified in the upper left corner.

Table 3
Performance comparison on remote sensing data with SIFT. The average RMSE, MAE, MEE, and RT are reported.

Method	RMSE	MAE	MEE	RT (ms)
RANSAC [24]	90.18	260.16	111.58	473.79
LPM [13]	1.34	22.49	**0.0004**	**47.36**
OANet++ [18]	23.48	77.04	25.71	67.21
CLNet [10]	12.55	55.44	12.57	74.39
MS^{2}DGNet [8]	4.58	32.89	3.38	85.73
PMA-Net (Ours)	**1.27**	**21.09**	**0.0004**	98.83

Table 4
The percentage of correctly estimated homomorphism, or accuracy (ACC.%) at different thresholds, where the thresholds are set to 3, 5, and 10 pixels.

Method	ACC.@3PX	ACC.@5PX	ACC.@10PX
PointCN [17]	67.93	82.59	92.76
OANet++ [18]	69.66	82.93	91.9
CLNet [10]	57.07	73.45	86.90
LMCNet [9]	72.93	83.62	92.76
MS^{2}DGNet [8]	72.07	83.28	92.62
PMA-Net (Ours)	**79.51**	**89.58**	**94.79**

the ablation results of PMA-Net for camera pose estimation. From Table 5, we can see that the FMP block, ADGC module, and MSP block sub-networks improve the performance in four evaluation indexes. Compared with the results of MS^{2}DGNet, combining the FMP block and ADGC module increases mAP5° by 9.52%. Similarly, combining the MSP block and ADGC module leads to mAP5° increase of 8.6%. Combining the FMP block, MSP block, and ADGC module outperforms MS^{2}DGNet by 11.06% for mAP5°. The increase of mAP5°, mAP10°, mAP15°, and mAP20° after combining the FMP block, MSP block, and the ADGC module shows that each module of our PMA-Net is valid.

4.5.2. Effect of the number of cascades on ADGC module

To investigate the impact of the number of cascades L in the ADGC module, evaluate the performance of the model trained on

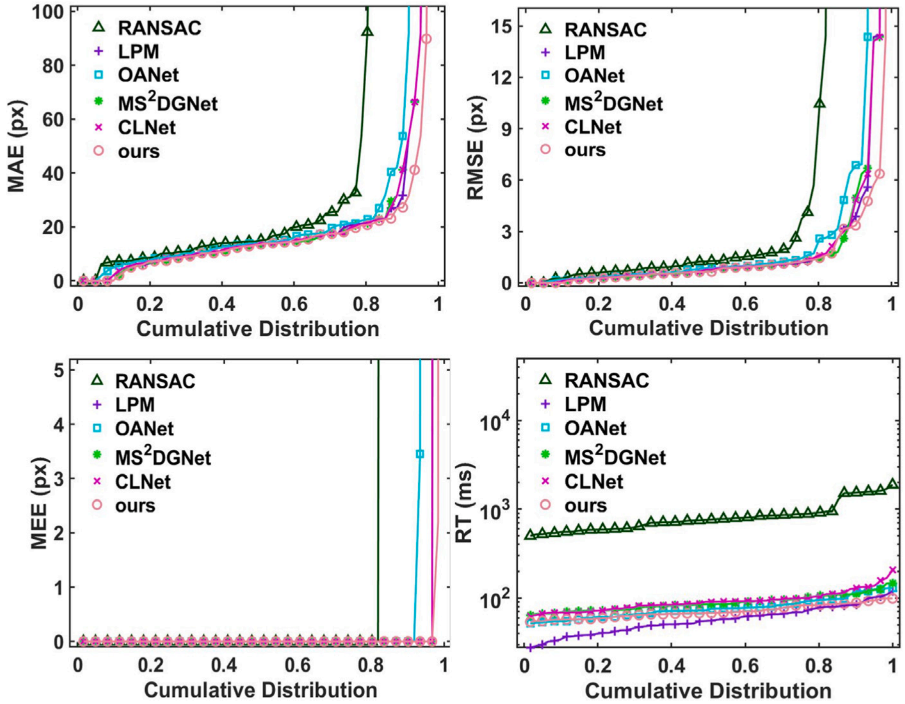

Fig. 7. Visualization of the image alignment cumulative distribution. The point coordinates (x, y) on this curve indicate that (100*x)% of the image pairs have performance values (i.e., RMSE, MAE, MEE, and RT) that do not exceed y.

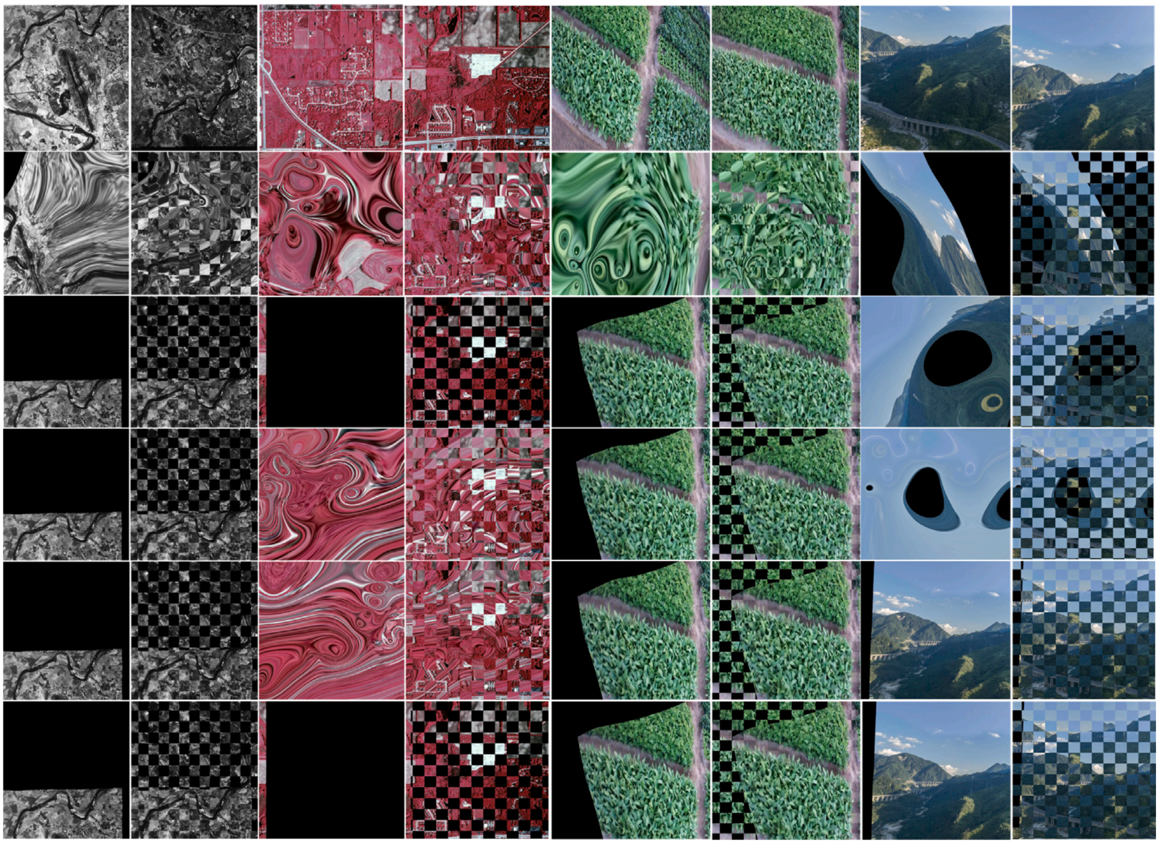

Fig. 8. Partial visualization results of remote sensing image registration. From left to right: SAR, CIAP, UAV and 720Yun images. The first row is the original image, and the following are the registration results of RANSAC, LPM, OANet, MS^{2}DGNet, and PMA-Net, respectively.

Table 5
Ablation studies on the YFCC100M dataset with SIFT descriptor. mAP5° (%), mAP10° (%), mAP15° (%) and mAP20° (%) for known scene without RANSAC are reported.

Baseline	FMP	ADGC	MSP	mAP5°	mAP10°	mAP15°	mAP20°
✓				38.36	51.08	58.77	64.04
✓	✓			41.39	53.27	60.66	65.78
✓		✓		44.91	56.62	63.57	68.33
✓			✓	40.78	52.79	60.01	65.18
✓	✓		✓	44.32	55.96	63.04	67.82
✓		✓	✓	46.96	57.94	64.66	69.05
✓	✓	✓		47.88	58.34	65.03	69.83
✓	✓	✓	✓	**49.42**	**59.97**	**67.97**	**71.86**

Table 6
Performance statistics of fusing global graph information in ADGC module with different number of cascades on YFCC100M dataset. The mAP5° (%), mAP10° (%), mAP15° (%) and mAP20° (%) for known scene without RANSAC are reported.

Number	mAP5°	mAP10°	mAP15°	mAP20°
$L = 0$	44.32	55.96	63.04	67.82
$L = 1$	47.98	59.16	65.92	70.49
$L = 2$	48.69	59.51	66.36	71.10
$L = 3$	**49.42**	**59.87**	**67.97**	**71.86**
$L = 4$	47.81	58.78	65.43	69.90

Table 7
Evaluation the performance of different methods on FMP block without RANSAC on the YFCC100M dataset for camera pose estimation. Size (MB): the number of network parameters.

Number	mAP5°	mAP10°	mAP15°	mAP20°	Size (MB)
PMA-Net+CA	40.65	53.02	60.17	65.43	3.25
Our PMA-Net	41.39	53.27	60.88	65.78	2.96

the YFCC100M dataset in the camera pose estimation experiment, as shown in Table 6. With an increase in the number of cascades L, PMA-Net gradually improves performance. This is attributed to the adaptive learning of global affinity between neighbor nodes in each stage through the dynamic cascading process, enabling a more comprehensive integration of global topological information. However, when the number of cascades reaches 4, there is a slight performance decline, indicating a certain level of overfitting in the network. Therefore, to balance efficiency and performance, the number of cascades L is set to 3.

4.5.3. Comparison scale factor in FMP block with other methods

To address the redundancy issue arising from low-dimensional mapping to high-dimensional channels, we design a channel branch within the FMP block. This channel branch leverages scaling factors and weight matrices to adaptively adjust the weights of different channel features, thereby reducing feature redundancy while maintaining stable performance. Here, we compare our proposed PMA-Net against PMA-Net with the FMP block using channel attention [50] in its second branch (PMA-Net+CA) for camera pose estimation. Results from Table 7 show that our PMA-Net perform better than PMA-Net+CA while using only 2.96M parameters, about 8.9% reduction with respect to PMA-Net+CA.

5. Conclusion

In this paper, we propose a novel network called PMA-Net to address the challenge of removing outliers from the consistency of inliers. Our PMA-Net progressively and adaptively learns the affinity between feature neighbors to effectively remove the random outliers embedded in the inlier consistency while overcoming redundant information introduced by mapping in the initial correspondences and the prediction of inlier probabilities. Extensive experiments conducted on various challenging tasks demonstrate the performance of PMA-Net, highlighting its effectiveness and generalization ability.

Although PAM-Net demonstrates competitive performance in eliminating mismatches and recovering camera poses, further improvements are still needed in the first branch of the FMP block, which utilizes spatial attention to explore the weights of different spatial features. We aim to design a lightweight method in the future that is as effective as the second branch in exploring the weights of different spatial features. We will further investigate this aspect in our subsequent work.

CRediT authorship contribution statement

Xiaojie Li: Writing – review & editing, Writing – original draft, Visualization, Software, Resources, Methodology, Conceptualization. **Fengyuan Zhuang:** Writing – original draft, Methodology, Formal analysis, Data curation. **Yizhang Liu:** Writing – review & editing, Writing – original draft, Project administration, Data curation. **Riqing Chen:** Writing – review & editing, Project administration, Funding acquisition, Conceptualization. **Lifang Wei:** Writing – review & editing, Methodology, Funding acquisition, Conceptualization. **Changcai Yang:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no conflict of interest.

Data availability

All the data used in this paper are public datasets.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under Grant 62171130, in part by the Natural Science Fund of Fujian Province under Grant 2023J01470, in part by the Fujian Innovation Fund Project under Grant 2021C0101, in part by the Fujian Pilot Project under Grant 2021H0013, in part by the Big Data in Agroforestry (Cross-Disciplinary) of Fujian Agriculture and Forestry University under Grant 712023030.

References

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

[2] R. Mur-Artal, J.M.M. Montiel, J.D. Tardos, ORB-SLAM: A versatile and accurate monocular SLAM system, IEEE Trans. Robot. 31 (5) (2015) 1147–1163.

[3] Y. Li, D. Miao, H. Zhang, J. Zhou, C. Zhao, Multi-granularity cross transformer network for person re-identification, Pattern Recognit. (2024) 110362.

[4] J. Ma, Y. Ma, C. Li, Infrared and visible image fusion methods and applications: A survey, Inf. Fusion 45 (2019) 153–178.

[5] L. David, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2004) 91–110.

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

[7] C. Zhao, Z. Cao, C. Li, X. Li, 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.

[8] L. Dai, Y. Liu, J. Ma, L. Wei, T. Lai, C. Yang, 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.

[9] Y. Liu, L. Liu, C. Lin, Z. Dong, 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.

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

[11] X. Liu, 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.

[12] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2003.
[13] J. Ma, J. Zhao, J. Jiang, H. Zhou, X. Guo, Locality preserving matching, Int. J. Comput. Vis. 127 (2019) 512–531.
[14] J. Ma, X. Jiang, J. Jiang, J. Zhao, X. Guo, LMR: Learning a two-class classifier for mismatch removal, IEEE Trans. Image Process. 28 (8) (2019) 4045–4059.
[15] Y. Xia, J. Ma, Locality-guided global-preserving optimization for robust feature matching, IEEE Trans. Image Process. 31 (2022) 5093–5108.
[16] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, et al., Graph attention networks, Stat 1050 (20) (2017) 10–48550.
[17] K.M. Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, P. Fua, Learning to find good correspondences, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2666–2674.
[18] J. Zhang, D. Sun, Z. Luo, A. Yao, L. Zhou, T. Shen, Y. Chen, L. Quan, 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.
[19] Y. Liu, Y. Li, L. Dai, C. Yang, R. Chen, Robust feature matching via advanced neighborhood topology consensus, Neurocomputing 421 (1) (2021) 273–284.
[20] X. Liu, G. Xiao, R. Chen, J. Ma, Pgfnet: Preference-guided filtering network for two-view correspondence learning, IEEE Trans. Image Process. 32 (2023) 1367–1378.
[21] J. Wang, X. Liu, L. Dai, J. Ma, L. Wei, C. Yang, R. Chen, PG-net: Progressive guidance network via robust contextual embedding for efficient point cloud registration, IEEE Trans. Geosci. Remote Sens. (2023).
[22] Y. Liu, B.N. Zhao, S. Zhao, L. Zhang, Progressive motion coherence for remote sensing image matching, IEEE Trans. Geosci. Remote Sens. 60 (2022) 1–13.
[23] Y. Liu, Y. Li, L. Dai, T. Lai, C. Yang, L. Wei, R. Chen, Motion consistency-based correspondence growing for remote sensing image matching, IEEE Geosci. Remote Sens. Lett. 19 (2021) 1–5.
[24] M.A. Fischler, R.C. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography - ScienceDirect, Read. Comput. Vis. (1987) 726–740.
[25] O. Chum, T. Werner, J. Matas, Two-view geometry estimation unaffected by a dominant plane, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'05, Vol. 1, IEEE, 2005, pp. 772–779.
[26] R. Raguram, O. Chum, M. Pollefeys, J. Matas, J.-M. Frahm, USAC: A universal framework for random sample consensus, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2012) 2022–2038.
[27] D. Barath, J. Noskova, J. Matas, Marginalizing sample consensus, IEEE Trans. Pattern Anal. Mach. Intell. 44 (11) (2021) 8420–8432.
[28] P.-E. Sarlin, D. DeTone, T. Malisiewicz, 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.
[29] X. Fan, L. Xing, J. Chen, S. Chen, H. Bai, L. Xing, C. Zhou, Y. Yang, VLSG-SANet: A feature matching algorithm for remote sensing image registration, Knowl.-Based Syst. 255 (2022) 109609.
[30] H. Xie, Y. Zhang, J. Qiu, X. Zhai, X. Liu, Y. Yang, S. Zhao, Y. Luo, J. Zhong, Semantics lead all: Towards unified image registration and fusion from a semantic perspective, Inf. Fusion 98 (2023) 101835.
[31] Z. Shi, G. Xiao, L. Zheng, J. Ma, R. Chen, JRA-Net: Joint representation attention network for correspondence learning, Pattern Recognit. 135 (2023) 109180.
[32] S. Zhang, J. Ma, ConvMatch: Rethinking network design for two-view correspondence learning, in: Proc. AAAI Conf. Artif. Intell, 2023, pp. 1–12.
[33] S. Pang, A. Du, M.A. Orgun, H. Chen, Weakly supervised learning for image keypoint matching using graph convolutional networks, Knowl.-Based Syst. 197 (2020) 105871.
[34] B. Thomee, D.A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, L.-J. Li, YFCC100M: The new data in multimedia research, Commun. ACM 59 (2) (2016) 64–73.
[35] W. Sun, W. Jiang, E. Trulls, A. Tagliasacchi, K.M. Yi, Acne: Attentive context normalization for robust permutation-equivariant learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11286–11295.
[36] Y. Cao, J. Xu, S. Lin, F. Wei, H. Hu, Gcnet: Non-local networks meet squeeze-excitation networks and beyond, in: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019.
[37] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, PMLR, 2015, pp. 448–456.
[38] C.R. Qi, L. Yi, H. Su, L.J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, Adv. Neural Inf. Process. Syst. 30 (2017) 5099–5108.
[39] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[40] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1492–1500.
[41] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[42] D. Kinga, J.B. Adam, et al., A method for stochastic optimization, in: International Conference on Learning Representations, ICLR, Vol. 5, San Diego, California;, 2015, p. 6.
[43] X. Jiang, J. Jiang, A. Fan, Z. Wang, J. Ma, Multiscale locality and rank preservation for robust feature matching of remote sensing images, IEEE Trans. Geosci. Remote Sens. 57 (9) (2019) 6462–6472.
[44] V. Balntas, K. Lenc, A. Vedaldi, K. Mikolajczyk, Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5173–5182.
[45] J. Xiao, A. Owens, 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.
[46] J. Zhang, D. Sun, Z. Luo, A. Yao, H. Chen, L. Zhou, T. Shen, Y. Chen, L. Quan, H. Liao, OANet: Learning two-view correspondences and geometry using order-aware network, IEEE Trans. Pattern Anal. Mach. Intell. 44 (6) (2022) 3110–3122.
[47] Z. Zhong, G. Xiao, L. Zheng, Y. Lu, 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.
[48] L. Zheng, G. Xiao, Z. Shi, S. Wang, J. Ma, MSA-Net: Establishing reliable correspondences by multiscale attention network, IEEE Trans. Image Process. 31 (2022) 4598–4608.
[49] X. Jiang, J. Ma, A. Fan, H. Xu, G. Lin, T. Lu, X. Tian, Robust feature matching for remote sensing image registration via linear adaptive filtering, IEEE Trans. Geosci. Remote Sens. 59 (2) (2020) 1577–1591.
[50] S. Woo, J. Park, J.-Y. Lee, I.S. Kweon, Cbam: Convolutional block attention module, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 3–19.