T-Net++: Effective Permutation-Equivariance Network for Two-View Correspondence Pruning

Guobao Xiao, *Senior Member, IEEE,* Xin Liu, Zhen Zhong, Xiaoqin Zhang, *Senior Member, IEEE,* Jiayi Ma, *Senior Member, IEEE,* and Haibin Ling, *Fellow, IEEE*

Abstract—We propose a conceptually novel, flexible, and effective framework (named T-Net++) for the task of two-view correspondence pruning. T-Net++ comprises two unique structures: the " $-$ " structure and the "|" structure. The " $-$ " structure utilizes an iterative learning strategy to process correspondences, while the "|" structure integrates all feature information of the " $-$ " structure and produces inlier weights. Moreover, within the "|" structure, we design a new Local-Global Attention Fusion module to fully exploit valuable information obtained from concatenating features through channel-wise and spatial-wise relationships. Furthermore, we develop a Channel-Spatial Squeeze-and-Excitation module, a modified network backbone that enhances the representation ability of important channels and correspondences through the squeeze-and-excitation operation. T-Net++ not only preserves the permutation-equivariance manner for correspondence pruning, but also gathers rich contextual information, thereby enhancing the effectiveness of the network. Experimental results demonstrate that T-Net++ outperforms other state-of-the-art correspondence pruning methods on various benchmarks and excels in two extended tasks. Our code will be available at https://github.com/guobaoxiao/T-Net.

Index Terms—Correspondence pruning, local-global, channel-spatial, permutation-equivariance

1 Introduction

Estimating correct feature point correspondences is indispensable for a variety of computer vision tasks, such as Structure from Motion (SfM), visual Simultaneous Localization and Mapping (SLAM), and image retrieval [1], [2]. Given a pair of images involving the same or similar scene, initial correspondences can be established by applying existing feature extraction methods, including traditional and learning-based methods [3], [4], [5]. However, they inevitably contain a large number of false correspondences ($i.e.$, outliers) besides correct correspondences ($i.e.$, inliers), especially for some difficult scenes ($e.g.$, large changes of viewpoint or illumination, occlusions, blur, and repetitive structures). This will severely hinder downstream tasks. Therefore, correspondence pruning as a key post-processing step has been adopted to remove outliers as much as possible.

During the past few decades, a number of correspondence pruning methods have been proposed. RANSAC [6] as one of the most popular methods and its variants, $e.g.$, [7], [8], [9], employ a hypothesize-and-verify strategy to search for a maximal set of inliers. Some non-parametric methods [10], [11], [12] analyze the relationship between correspondences to remove outliers. Nevertheless, these methods often fail to achieve satisfying results when initial correspondences contain a large proportion of outliers.

Some learning-based methods have received much attention for correspondence pruning due to the powerful learning and representation ability of deep learning. LFGC [13] is the first one to formulate the correspondence pruning as a classification problem and an essential matrix regression problem by using a deep network. Note that, the input correspondences are usually unordered, and this requires the network to be permutation-equivariance. That is, the network should not be influenced by the order of input, and this makes convolutional layers not applicable. Then LFGC adopts Multi-Layer Perceptrons (MLPs) to process correspondences independently, however, simple MLPs encounter challenges in effectively capturing rich contextual information.

To improve the performance of networks, some works have been proposed by designing different network structures and learning paradigms. For example, NM-Net [14] and LMCNet [15] exploits the neighborhood information to capture contextual information; OANet [16] uses differential pooling and unpooling layers to achieve the local context; ACNe [17], CSR-Net [18], PESA-Net [19] and MS^{2}DG-Net [20] design different attention mechanisms to explore rich contextual information. These networks are able to improve the performance over LFGC, however, there exist some limitations. On one hand, most of them employ an iterative network structure, which takes the inlier weights and residuals of the previous iteration sub-network as additional inputs of the following iteration sub-network, to improve the performance. But, this simple iterative strategy cannot fully exploit the feature information from the previous iteration sub-network, which results in the waste of important feature information and sub-optimal performance. On the other hand, they adopt a PointNet-like architecture (denoted as the PointCN module), which consists of some normalization layers and MLPs layers, as the network backbone to process the correspondence data. While, a solitary sequential structure often faces challenges in capturing potential relationships between correspondences, $e.g.$, the consistency of inliers. Thus, optimal network structures and backbones are essential for maximizing the performance of correspondence pruning.

To address the above problems, we first propose a novel and effective network (called T-Net++) to comprehensively exploit the feature information of all iteration sub-networks. As shown in

- *Guobao Xiao, Xin Liu and Zhen Zhong are with School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, 310018, China.*
- *Xiaoqin Zhang is with College of Computer Science and Artifical, Wenzhou University, Wenzhou, China, 350108.*
- *Jiayi Ma is with Electronic Information School, Wuhan University, Wuhan, 430072, China.*
- *Haibin Ling are with Department of Computer Science, Stony Brook University, Stony Brook, NY 11794, USA (e-mail: hling@cs.stonybrook.edu).*

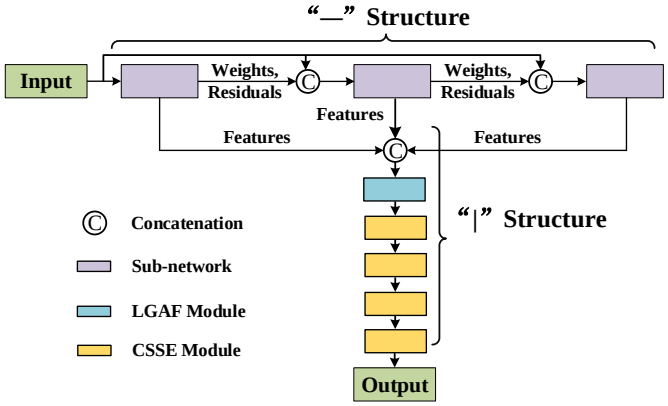


Fig. 1. The proposed T-Net++ architecture. It consists of two main parts: the " – " structure and the "|" structure.

Fig. 1, T-Net++ mainly contains two key parts: the " – " and "|" structures. It starts with the " – " structure, which consists of three iteration sub-networks composed by some learning modules to process the correspondence data. To increase the diversity of input information for each iteration sub-network, the inlier weights and residuals calculated from the previous sub-network serve as additional inputs of the following iteration sub-network. Next, we introduce a "|" structure for the integration of features from all iteration sub-networks. Within this structure, we develop a Local-Global Attention Fusion (LGAF) module to harness the combined feature information from the local and global attention branches. Specifically, the local attention branch focuses on point-wise attention to enhance the representation of individual correspondences, while the global attention branch employs two pooling operations in the spatial dimension to determine the channel-wise importance of different features. Following this, the "|" structure, with the assistance of four learning modules, produces the inlier probability for each correspondence.

In addition, we develop a Channel-Spatial Squeeze-and-Excitation (CSSE) module as our network backbone to replace the original PointCN module. The proposed CSSE module consists of a series of squeeze and excitation operations. The squeeze operation is used to improve the global receptive field of features and reduce the computational overhead, and the excitation operation is used to learn the potential and complex relationships of features. Specifically, we start by compressing the spatial dimensions of features in two distinct ways to extract potential channel relationships through our excitation operation. Afterward, we squeeze the channel dimension of features to capture potential correspondence relationships using the same excitation operation. These two operations are then integrated into the center of the PointCN module to form the CSSE module.

In summary, the main contributions of this work are as follows:

- We present T-Net++, a straightforward yet remarkably efficient network incorporating two vital structures: the " – " and "|" structures. These structures are skillfully engineered to fully harness valuable feature information from all iteration sub-networks. Moreover, our network incorporates a Local-Global Attentional Fusion (LGAF) module to enhance the utilization of fused features.
- We develop an innovative network backbone known as the Channel-Spatial Squeeze-and-Excitation (CSSE) module. This module adeptly captures complex relationships among features through the application of squeeze-and-excitation operations along various dimensions. It amplifies the representation capabilities of essential channels and correspondences.
- Our proposed T-Net++ excels in the precise identification of inliers and the accurate recovery of camera poses between two matching images. Its outstanding performance improvements over current state-of-the-art methods are substantiated by both qualitative and quantitative results across multiple datasets and benchmarks.

This paper is an extension of our previous work (T-Net) [21]. We have made several crucial improvements, including a novel Local-Global Attentional Fusion (LGAF) module to improve the effectiveness of concatenated features, an enhanced Spatial-Channel Squeeze-and-Excitation (CSSE) module to enhance the representation ability of features, more theoretical analyses and more experimental analysis to evaluate the performance of our method. We have also added two new extended tasks, i.e., image registration and point cloud registration. In T-Net, we previously concatenate the learned features from three iteration sub-networks exclusively along the channel dimension, thereby restricting the representation capacity of the fused features. Simultaneously, it becomes evident that these three features, acquired from different sub-networks, could possess varying degrees of importance for the final correspondence learning, as confirmed by the ablation study. In T-Net++, the newly introduced LGAF module offers a more comprehensive approach to harness the fused feature from two distinct perspectives. Moreover, in T-Net, we primarily focus on the channel-level response via averaging operations, limiting our ability to capture potential relationships among correspondences. Our newly proposed CSSE module is specifically designed to capture potential relationships within feature maps, taking into account both channel and spatial dimensions in unique ways.

The remaining paper is organized as follows: In Section 2, we provide an overview of the related literatures. We formulate our two-view correspondence pruning task and the proposed T-Net++ in Section 3. In Section 4, we present the experimental results on different datasets and benchmarks. We draw conclusions in Section 5.

2 Related Work

In this section, we briefly review the related work on correspondence pruning, including handcrafted and learning-based methods.

2.1 Handcrafted Correspondence Pruning Methods

The most representative handcrafted methods for correspondence pruning are RANSAC [6] and its variations over the past decades. RANSAC designs a hypothesize-and-verify framework to iteratively generate parametric model hypotheses and verify its reliability. Specifically, it randomly samples a minimal subset of data to get a model hypothesis and then verifies its confidence by the number of consistent correspondences. Based on this framework, numerous works adopt different strategies to improve the accuracy and efficiency of the plain RANSAC. For example, [7] finds dominant model instances progressively via a RANSAC-like sampling. MAGSAC [22] designs a weighted least-squares fitting to obtain the optimized model and improves the model quality by applying sigma-consensus. Graph-Gut RANSAC [23] introduces graph-cut to RANSAC for obtaining optimized solutions. D2Fitting [8]

incrementally explores dominant instances by combing model generation and selection. MSHF [9] uses hypergraph modeling to capture multiple-structure fitting problems. This type of methods are still regarded as a standard solution for robust parameter model estimation. However, these methods have grave uncertainty, because their performance relies on the sampled subset. When the number of outliers in the initial correspondences established by existing feature extraction methods is large, they cannot achieve satisfactory results.

Furthermore, some non-parametric methods have been proposed to select inliers by the specific geometric constraint of correspondences. VFC [24] introduces a vector field consensus for non-rigid point matching. GMS [10] uses a simple and efficient grid-based means to convert the motion smoothness constraints for removing outliers. LPM [25], RFM-SCAN [26], and COMR [12] select inliers by evaluating the consistency of local neighborhood structures. Although these methods have shown impressive results in terms of efficiency and accuracy, they cannot address the problem that initial correspondences contain a large proportion of outliers, because their performance will gradually decrease or even fail with the increase of outliers.

2.2 Learning-based Correspondence Pruning Methods

With the flourishing of deep learning, learning-based feature matching methods have recently advanced substantially on performance. To extract more accurate feature keypoints and distinguish descriptors than traditional feature extraction methods, many works [4], [5] use Convolutional Neural Network (CNN) to process images, and obtain decent performance. However, they inevitably generate a large number of outliers, especially for some challenging matching scenarios, which are also in great need of correspondence pruning as the post-processing.

As a new direction, DSAC [27] designs a differentiable counterpart of RANSAC by soft argmax and probabilistic selection, but it still has a great deal of uncertainty. Inspired by the success of PointNet [28], [29] for 3D point cloud classification and segmentation, LFGC [13] first attempts to represent the outliers removal task as a correspondence classification problem and an essential matrix regression problem. It adopts a PointNet-like architecture, which is based on MLPs to keep permutation-equivariance of inputs, to label the correspondences as inliers or outliers. It also designs a Context Normalization (CN) to capture global context and a weighted eight-point algorithm to regress the essential matrix. Subsequently, DFE [30] also utilizes PointNet-like architecture and CN to process correspondences, but adopts a different loss function and an iterative strategy to estimate the fundamental matrix. NM-Net [14] presents a hierarchical network to capture reliable local context by compatibility-specific mining method for searching more consistent neighbors than the spatially k-nearest neighbors of correspondences. OANet [16] adopts a clustering method to exploit both local and global spatial contexts, and designs a simple iterative way to achieve significant performance improvements. ACNe [17] develops a simple yet effective Attentive Context Normalization to replace CN proposed by LFGC for reliable contextual information. MS^{2}DG-Net [20] builds sparse semantic graphs to capture the local topology among correspondences, and it also uses the same iterative way of OANet.

Compared with existing learning-based methods, the proposed method in this paper not only preserves the permutation-equivariance property for correspondence pruning, but also gathers rich contextual information to improve the effectiveness of network, by designing a novel T-Structure and two novel modules.

3 Method

In this paper, the correspondence pruning problem is formulated as a correspondence classification problem ($i.e.$, inlier or outlier) and an essential matrix regression problem. In this section, we describe the details of our proposed T-Net++. Specifically, we first describe the problem formulation of two-view correspondence pruning task in Section 3.1. Then, we develop a T-Structure network to integrate the information of each sub-network in Section 3.2. Next, in Section 3.3, we propose a novel Channel-Spatial Squeeze-and-Excitation module for better learning. Finally, we introduce our network architecture and loss function in Section 3.4 and Section 3.5, respectively.

3.1 Problem Formulation

For the given two images (I, I') with the same scene, we first employ existing feature extraction methods, such as SIFT [3] or SuperPoint [5], to extract image keypoints and construct corresponding descriptors. Then, we create a putative feature point correspondence set S, which includes N correspondences, according to the nearest neighbor search strategy of descriptors:

$$S = \{s_1, s_2, ..., s_N\} \in \mathbb{R}^{N\times 4}, s_i = (u_i, v_i, u_i^{'}, v_i^{'}), \quad (1)$$

where s_i indicates the i-th putative correspondence, and (u_i, v_i) and $(u_i^{'}, v_i^{'})$ are two keypoint coordinates, which have been normalized by camera intrinsics, on the matching pair (I, I'), respectively. The full network structure is illustrated in Fig. 1. In particular, our T-Net++ takes the putative feature point correspondence set S as network input, and finally produces the corresponding weight set W:

$$W = \{w_1, w_2, ..., w_N\}, w_i \in [0, 1), \quad (2)$$

where w_i denotes the probability of putative correspondences s_i being an inlier. $w_i > 0$ indicates that the putative correspondence s_i is an inlier, and vice versa. Then, we use a weighted eight-point algorithm designed by LFGC [13] to regress the essential matrix, which is used for recovering the camera pose ($i.e.$, corresponding rotation and translation vectors) of two matching images. The weighted eight-point algorithm is more robust than the traditional eight-point algorithm [31], because it reduces the negative influence of outliers by utilizing the weight set W. Therefore, the whole process can be written as:

$$W = \tanh(\mathrm{ReLU}(o)), o = f_\theta(S), \quad (3)$$

$$\hat{E} = g(S, W), \quad (4)$$

where $f_\theta(\cdot)$ indicates our permutation-equivariance neural network with the learned network parameters θ, and o is the logit values for correspondence classification. Activation function $\tanh(\cdot)$ and $\mathrm{ReLU}(\cdot)$ are used to compute inlier weights W. $g(\cdot, \cdot)$ represents the weighted eight-point algorithm to compute the essential matrix $\hat{E}$ by self-adjoint eigen-decomposition.

3.2 T-Structure Network

In previous works [30], [16], [20], an iterative network structure is adopted to promote the network performance for correspondence pruning. However, these methods utilize only final outputs of

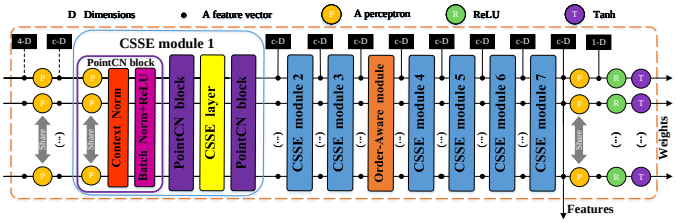


Fig. 2. Schematic diagram of the proposed sub-network. It mainly contains seven CSSE modules and one Order-Aware module.

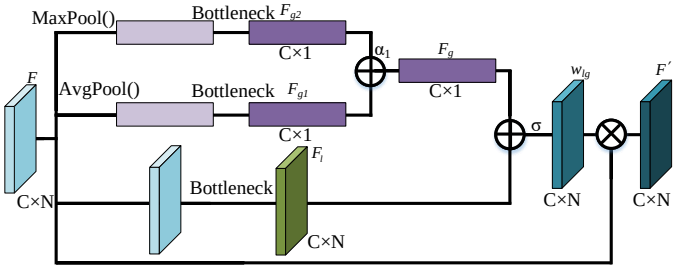


Fig. 3. The structure diagram of our proposed LGAF module.

the previous iteration sub-network as additional inputs of the following iteration sub-network, while a large portion of feature information in the previous iteration sub-network cannot be fully exploited. This operation will cause the waste of much important information and hinder network performance for correspondence pruning.

To address this problem, we first develop a novel structure (called T-Structure). As shown in Fig. 1, T-Structure includes two parts: the " – " structure and the "|" structure. Our " – " structure is composed of a number of iteration networks (called sub-networks in this paper) for two-view correspondence pruning. As shown in Fig. 2, each sub-network consists of some Channel-Spatial Squeeze-and-Excitation (CSSE) modules (see Sec. 3.3) and an Order-Aware module proposed by OANet [16]. Meanwhile, in order to improve the diversity of input information for network learning, each sub-network will not only take the putative feature point correspondence set as input, but also add the inlier weights and residuals predicted by the previous sub-network as the additional inputs. Our "|" structure is used to integrate the feature information of all sub-networks and output the final inlier probability of each putative correspondence by several learning modules. The "|" structure includes three novel operations: feature extraction, feature fusion, and feature learning.

Feature Extraction: To adequately exploit the valuable feature information of all sub-networks, our "|" structure first extracts the output features of each sub-network. Here, as [21], we first adopt a simple concatenation operation to obtain a new feature:

$$F = \mathrm{Concat}(\mathrm{F}_1, \mathrm{F}_2, ..., \mathrm{F}_\mathrm{i}) \tag{5}$$

where $\mathrm{Concat}(\cdot)$ represents the concatenation operation along the channel dimension of features. $F_i \in \mathbb{R}^{c\times N}$ denotes the output feature of the i-th sub-network, and c is the channel number and N is the correspondence number. $F \in \mathbb{R}^{C\times N}$ is the concatenation feature, where $C = i * c$.

Feature Fusion: Although it is possible to extract the feature information of all sub-networks by using the concatenation operation, this simple operation cannot integrate all features adequately. Meanwhile, as described in Sec. 1, the focus of features learned from each sub-network is varied. Therefore, we develop a novel Local-Global Attention Fusion (LGAF) module, as shown in Fig. 3, to better fuse feature information. We design a three-branch structure to explore point-wise and channel-wise attention of the concatenation feature F by the local and global operations. Specifically, for the concatenation feature $F = \in \mathbb{R}^{C\times N}$, its point-wise feature can be obtained by a local operation as follows:

$$F_l = \mathrm{BN}(\mathrm{MLP}_2(\mathrm{ReLU}(\mathrm{BN}(\mathrm{MLP}_1(\mathrm{F}))))). \tag{6}$$

Here, we utilize a bottleneck structure to reduce the computational overhead. $\mathrm{MLP}_1(\cdot)$ is a dimension reduction MLP layer with the size of $\mathbb{R}^{C\times\frac{C}{r}}$ and $\mathrm{MLP}_2(\cdot)$ is a dimension increasing MLP layer with the size of $\mathbb{R}^{\frac{C}{r}\times C}$, where r represents the channel reduction ratio. The MLP layer with the 1×1 kernel size has been adopted to capture the local context of correspondences. $\mathrm{BN}(\cdot)$ denotes the Batch Normalization layer. $F_l \in \mathbb{R}^{C\times N}$ is the local context feature of the F. It aims at exploring point-wise context to enhance the representation ability of each correspondence.

Then, we use a two-branch structure to get the channel-wise feature by two global operations of the concatenation feature:

$$F_{g1} = \mathrm{BN}(\mathrm{MLP}_4(\mathrm{ReLU}(\mathrm{BN}(\mathrm{MLP}_3(\mathrm{AvgPool}(\mathrm{F})))))), \tag{7}$$

$$F_{g2} = \mathrm{BN}(\mathrm{MLP}_6(\mathrm{ReLU}(\mathrm{BN}(\mathrm{MLP}_5(\mathrm{MaxPool}(\mathrm{F})))))), \tag{8}$$

where $\mathrm{AvgPool}(\cdot)$ and $\mathrm{MaxPool}(\cdot)$ are the global average-pooling layer and the global max-pooling layer in the spatial dimension of features, respectively, to capture the global context feature from different aspects. $F_{g1} \in \mathbb{R}^{C\times 1}$ denotes the global average context feature of the F, and $F_{g2} \in \mathbb{R}^{C\times 1}$ represents the global maximal context feature of the F. They aim at capturing channel-wise context to obtain the importance of different features. We also adopt the same bottleneck structure with the local context to avoid excessive overhead. Noteworthily, these MLP layers have independent network parameters for network learning.

Next, we integrate the two types of global contexts:

$$F_g = F_{g1} + \alpha_1 F_{g2}, \tag{9}$$

where $F_g \in \mathbb{R}^{C\times 1}$ is the global context feature of the F. α_1 is a learned parameter to balance the two terms. Meanwhile, we use a spread function to ensure that the F_g has the same shape as the local context F_l. Then, the local-global attention weight $w_{lg} \in \mathbb{R}^{C\times N}$ can be obtained as:

$$w_{lg} = \sigma(F_l + F_g), \tag{10}$$

where σ represents the Sigmoid activation function. w_{lg} is the local-global attention weight generated by LGAF module, which can enhance the representation ability of important channels and correspondences. Finally, we get the refined feature F':

$$F' = w_{lg} \otimes F, \tag{11}$$

where $\otimes$ denotes the element-wise multiplication operation. F' is the final output of LGAF module, which has strong representation ability by exploring the point-wise and channel-wise attention of the F.

Compared to T-net [21], our designed LGAF module in T-net++ has two key advantages. Firstly, it fully leverages the useful information of the concatenation feature from local and global aspects to obtain point-wise and channel-wise attention for enhancing the important elements. Secondly, it only involves some

simple pooling and MLPs operations with reasonable computational overhead.

Feature Learning: After the feature fusion operation, we use four CSSE modules for feature learning:

$$F_{final} = f_{|}(F'), \tag{12}$$

where F_{final} is the final output of our "|" structure. $f_{|}$ represents our "|" structure, which contains four CSSE modules. Therefore, we can obtain the final inlier weight $W = \tanh(ReLU(F_{final}))$.

3.3 CSSE Module

In this paper, the sub-network is used for processing correspondence features, which is crucial for correspondence pruning. However, previous works [13], [14], [16], [17] use the PointCN module as the network backbone. PointCN mainly contains two identical sequential blocks (denoted as PointCN block): an MLP layer to process unordered and irregular correspondences, a Context Normalization layer to capture the global context, and a Batch Normalization layer with a ReLU activation function to accelerate network training. This single sequential structure utilizes indiscriminate MLPs and normalization layers for network learning, which may hinder the performance of network since the inputs of our network contain a large number of outliers. Therefore, to fill this gap, as illustrated in Fig. 2, we develop a novel Channel-Spatial Squeeze-and-Excitation (CSSE) module as network backbone. We use a CSSE layer to capture the potential relationship among features by a series of squeeze-and-excitation operations in the channel and spatial dimensions.

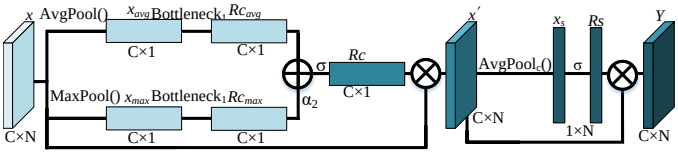


Fig. 4. The structure diagram of our proposed CSSE layer.

Our CSSE layer is shown in Fig. 4. To be specific, for an intermediate feature $x \in \mathbb{R}^{c\times N}$, we first adopt two different squeeze operations in the spatial dimension of the x to aggregate channel information:

$$x_{avg} = \mathrm{AvgPool(x)}, \tag{13}$$

$$x_{max} = \mathrm{MaxPool(x)}, \tag{14}$$

where $x_{avg} \in \mathbb{R}^{c\times 1}$ and $x_{max} \in \mathbb{R}^{c\times 1}$ are the average channel feature and the maximum channel feature, respectively. Next, we use the excitation operation to learn the potential and complex relationship cues of channels:

$$Rc_{avg} = \mathrm{FC_2(ReLU(FC_1(x_{avg})))}, \tag{15}$$

$$Rc_{max} = \mathrm{FC_4(ReLU(FC_3(x_{max})))}, \tag{16}$$

where $\mathrm{FC_i}(\cdot)$ represents the fully connected layer. We also utilize the bottleneck structure to reduce the parameter overhead. Rc_{avg} and Rc_{max} are the average channel relationship feature and maximum channel relationship feature, respectively. After that, we can obtain the channel relationship Rc:

$$Rc = \sigma(Rc_{avg} + \alpha_2 Rc_{max}), \tag{17}$$

where α_2 is a learned hyper-parameter to balance the two terms. The channel relationship $Rc \in \mathbb{R}^{c\times 1}$ indicates the importance of channels of the intermediate feature x. Therefore, Rc is used for enhancing the representation ability of important channels:

$$x' = Rc \otimes x, \tag{18}$$

where x' is the feature enhanced by channel relationship Rc. Then, we utilize the squeeze operation in the channel dimension of the feature x' to aggregate spatial information:

$$x_s = \mathrm{AvgPool_c(x')}. \tag{19}$$

Here, we only adopt global average-pooling in the channel dimension $AvgPool_c(\cdot)$ to squeeze features. $x_s \in \mathbb{R}^{1\times N}$ is average spatial feature feature. Next, we directly get the spatial relationship Rs:

$$Rs = \sigma(x_s), \tag{20}$$

The spatial relationship $Rc \in \mathbb{R}^{1\times N\times 1}$ indicates the importance of correspondences of the feature x'. Therefore, we use Rc to enhance the representation ability of potential inliers:

$$Y = Rs \otimes x', \tag{21}$$

where Y is the final output feature of our CSSE layer.

It is worth pointing out that, in our T-Net [21], we propose a Permutation-Equivariance Context Squeeze-and-Excitation module as network backbone to replace PointCN module. But, it focuses only on the channel-wise relationship using the averaging operation, which is limited and ignores the potential relationship of other features ($e.g.$, correspondences). Our CSSE module effectively refines intermediate features by sequentially enhancing and suppressing features from both channel and spatial dimensions. Furthermore, our module is lightweight.

3.4 Network Architecture

As shown in Fig. 1, our T-Net++ mainly contains two structures: the " $-$ " structure and the "|" structure. The " $-$ " structure includes three sub-networks, which are used to process the correspondence data. The "|" structure is mainly used to integrate the feature information from all sub-networks and output the final inlier weights, which consists of an LGAF module and four CSSE modules.

The structure of the sub-network is shown in Fig. 2. It contains seven CSSE modules and an Order-Aware module. The details of CSSE module have been introduced in Section 3.3. The Order-Aware module is proposed by OANet [16] to effectively capture local and global contexts. In each sub-network of T-Net++, we insert an Order-Aware module to improve the network capacity for information processing. Specifically, the Order-Aware module includes three different parts: a Differentiable Pooling layer, an Order-Aware Filtering block, and a Differentiable Unpooling layer. The Differentiable Pooling layer learns a soft assignment matrix to map N correspondences into M clusters for exploring the local context. The Order-Aware Filtering block transforms the channel of spatial dimensions of clusters and utilizes PointCN block to capture the global context of clusters. The Differentiable Unpooling layer also leans a soft assignment matrix to recover the original order and size of correspondences. In each sub-network, we utilize a Differentiable Pooling layer, three Order-Aware Filtering blocks, and a Differentiable Unpooling layer for network learning.

The permutation-equivariance property indicates that the input and output of network always match with each other no matter how the order of the input is changed. This property is very important for our correspondence pruning network since the inputs are unordered and irregular for feature matching. We discuss the permutation-equivariance of T-Net++, which mainly contains the CSSE, LGAF, and Order-Aware modules. The permutation-equivariance of Order-Aware module has been proved in [16]. Therefore, in this paper, we prove the permutation-equivariance of CSSE and LGAF modules.

The proposed CSSE and LGAF modules are insensitive to the order of input features, since the pooling operation, MLP, and the fully connected layers used in our method are invariant for input. Here we summarize a theoretical explanation. Given an input feature $\mathbf{X} \in \mathbb{R}^{C\times N}$ (we omit the last dimension for simplicity), we impose a permutation transformation $\mathbf{P}$ on it. For the average-pooling and operation and max-pooling operation, the permutation matrix $\mathbf{P}$ only changes the order of elements $\{x_\xi\}_{\xi=1}^{N}$ in $\mathbf{X}$, but it does not change their values. Obviously, they have no effect on the calculation of average and maximal values:

$$\mathrm{Avepool}(\mathbf{PX}) = \frac{1}{\mathrm{N}}\sum_{\xi=1}^{\mathrm{N}} \mathrm{x}_\xi = \mathrm{AvgPool}(\mathbf{X}). \tag{22}$$

$$\mathrm{Maxpool}(\mathbf{PX}) = \underset{\mathrm{x_i}\in\mathrm{X}}{\mathrm{Max}}(\mathrm{x_i}) = \mathrm{MaxPool}(\mathbf{X}). \tag{23}$$

For an MLP layer, it can be present to:

$$\mathrm{MLP}(\mathbf{X}) = \mathbf{X}\mathbf{W}' \oplus \mathbf{B}, \tag{24}$$

$$\mathrm{MLP}(\mathbf{PX}) = \mathbf{PX}\mathbf{W}' \oplus \mathbf{B} = \mathbf{P}\mathrm{MLP}(\mathbf{X}), \tag{25}$$

where $W' \in R^{C\times C}$ and $B \in R^{C\times 1}$ are a learned weight matrix and a bias vector of an MLP layer, respectively. $\oplus$ means the element-wise summation. Note that, B is added to each correspondence, and it will not affect permutation-equivariance. The proof of the fully connected layer is the same as the MLP layer. Our T-Net++ mainly includes several permutation-equivariance modules, which makes our network not sensitive to the order of input correspondences.

3.5 Loss Function

Following previous work in [13], [16], [17], we adopt a hybrid loss function to optimize our neural network as follows:

$$\mathcal{L} = \mathcal{L}_c(W, L) + \beta\mathcal{L}_e(E, \hat{E}), \tag{26}$$

where β is a weight to balance the two loss functions. $\mathcal{L}_c(\cdot)$ is a binary cross entropy loss ($i.e.$, classification loss):

$$\mathcal{L}_c(W, L) = \frac{1}{N}\sum_{i=1}^{N}\gamma_i H(w_i, L_i), \tag{27}$$

where γ_i represents the per-label weight for balancing the outlier/inlier ratio, which has been set to 0.5 for simplicity. $H(\cdot)$ is the binary cross entropy function. We utilize the geometric distance of correspondence to determine the ground-truth label L for each correspondence:

$$d(s_i, E) = \frac{(q_i^{'T} E q_i)^2}{\|Eq_i\|^2_{[1]} + \|Eq_i\|^2_{[2]} + \|E^T q_i'\|^2_{[1]} + \|E^T q_i'\|^2_{[2]}}, \tag{28}$$

where $s_i = (q_i, q_i')$ is a putative correspondence, which contains two keypoint coordinates (q_i and q_i'). $A_{[i]}$ represents the i-th item of vector A. $d(\cdot)$ denotes geometric distance of correspondence. We set 10^{-4} as the label threshold.

$\mathcal{L}_e(\cdot)$ is a geometry loss ($i.e.$, essential matrix loss) based on the above geometric distance formula:

$$\mathcal{L}_e(E, \hat{E}) = \frac{(p^{'T}\hat{E}p)^2}{\|Ep\|^2_{[1]} + \|Ep\|^2_{[2]} + \|E^T p'\|^2_{[1]} + \|E^T p'\|^2_{[2]}}, \tag{29}$$

where p and p' denote two virtual keypoint coordinates, and they are generated by using the ground truth essential matrix E.

4 Experiments

In this section, we evaluate the network performance of our method and compare it with some state-of-the-art methods on large public datasets. We use the correspondence classification and camera pose estimation tasks to demonstrate the effectiveness of our correspondence pruning network. Meanwhile, we extend our method to remote sensing image registration and point cloud registration tasks to verify the generalization ability. In the following, we first introduce the details of the experiment, including datasets, evaluation metrics, and implementation details, then report the comparative results and analyze the ablation study. Finally, we show the results of two extended tasks.

4.1 Datasets

Outdoor Scenes. We select the Yahoo's YFCC100M dataset [32] as outdoor scenes. It is a tourist image collection from the Internet, which has about 100 million photos. Following previous OANet [16], the entire dataset is split into 71 image sequences based on different tourist spots. We utilize 67 image sequences as training data, and the remaining 4 image sequences (i.e., Buckingham Palace, Sacre Coeur, Reichstag, and Notre Dame Front Facade) as unknown scenes to test the generalization ability of network.

Indoor Scenes. We select the SUN3D dataset [33] as indoor scenes. It is an image frame collection obtained from some RGBD videos. The entire dataset is split into 254 image sequences. We utilize 239 image sequences as training data, and the remaining 15 image sequences as unknown scenes to test the generalization ability of network.

In this paper, we use both known scenes and unknown scenes to test network generalization ability. The unknown sequences are the above testing image sequences ($i.e.$, the 4 image sequences and 15 image sequences). To obtain the known scenes, we split the above training image sequences into three disjoint subsets, $i.e.$, the training set (60%), validation set (20%), and testing set (20%). For each unknown image sequence, we randomly generate $1,000$ matching image pairs for testing. For each known image sequence, we randomly generate $10,000$ matching image pairs for training and generate 100 matching image pairs for validation and testing. Furthermore, we re-train all comparative models under the same setting and datasets for fair comparison.

4.2 Implementation Details

In this paper, we utilize off-the-shelf feature extraction methods, such as traditional SIFT [3], to establish the $N \times 4$ putative feature point correspondence set S as the input of the neural network. The average number N of putative correspondences is about 2000. The channel dimensions of all correspondence features are 128 besides our LGAF module. Our three sub-networks all have 7 CSSE modules and one Order-Aware module. For the

Order-Aware module, it maps N correspondences into M clusters (typically $M = 500$). Meanwhile, we keep the same setting with OANet [16], $i.e.$, a Differentiable Pooling layer, 3 Order-Aware Filtering blocks, and a Differentiable Unpooling layer. For the "|" structure, it uses one LGAF module and 4 CSSE modules to predict the final inlier weights. The learned parameter α_1 in the LGAF module and α_2 in the CSSE module are set as 1. Our network is implemented by using Pytorch. We adopt the Adam optimizer with the learning rate of 10^{-3} and batchsize of 32 to train the network. The parameter β in the hybrid loss function is set to 0 for the first 20k iterations and then fixed to 0.5 for the remaining 480k iterations.

4.3 Evaluation Metrics

We use the accuracy of camera pose estimation to evaluate the performance of two-view correspondence pruning. The camera poses of two matching images, $i.e.$, corresponding rotation and translation vectors implemented by the OpenCV function, can be recovered from the predicted essential matrix. The angular differences between the recovered and the ground-truth rotation/translation vectors are adopted as the error metrics. We compute mean average precision (mAP) of the angular differences at different thresholds as the evaluation criterion. Moreover, we use RANSAC with a 0.001 threshold as post-processing, which takes the estimated inliers of networks as input, to further improve the performance of camera pose estimation.

Furthermore, we report $Precision(P)$, $Recall(R)$, and $F\text{-}score(F)$ as the evaluation metrics to evaluate the effectiveness of correspondence classification. Specifically, P is defined as the number ratio between the identified correct feature point correspondences and the reserved correspondences. R is defined as the ratio of the identified correct feature point correspondences and the actual correspondences contained in the putative feature point correspondence set. F is calculated as $2 * P * R/(P + R)$.

4.4 Comparative Results

We compare our method with some state-of-the-art methods [6], [22], [25], [29], [30], [13], [16], [17], [34], [21]. For traditional RANSAC [6] and MAGSAC [22], we adopt the additional ratio test during the feature descriptor matching [3] to improve their performance. For implementing these learning-based methods, we directly utilize the released code and train them under the same training set. PointNet++ [29] is an improved version of PointNet [28], which improves network performance by capturing local context of point sets. We adopt a 4D Euclidean space ($i.e.$, coordinate space) to search the neighbors of each correspondence. DFE [30] is a robust concurrent method to estimate the fundamental matrix, and we use the essential matrix to replace the fundamental matrix for pose estimation. LFGC [13] is a pioneering learning-based correspondence classification work, and we use our hybrid loss for better performance. ACNe [17] designs a learned attentive context normalization to capture local and global contexts, we implement it by using a Pytorch version. OANet [16] presents a cluster operation to obtain local context, and we employ an official iterative version ($i.e.$, OANet++) for comparison. MS^{2}DG-Net [20] builds sparse semantics graphs based on the semantics similarity of correspondences to capture local context, and we directly quote the results of the paper as the code is not available at the moment. SuperGlue [34] is a state-of-the-art graph neural network to match two sets of local features, and we use the official implementation to recover transformation.

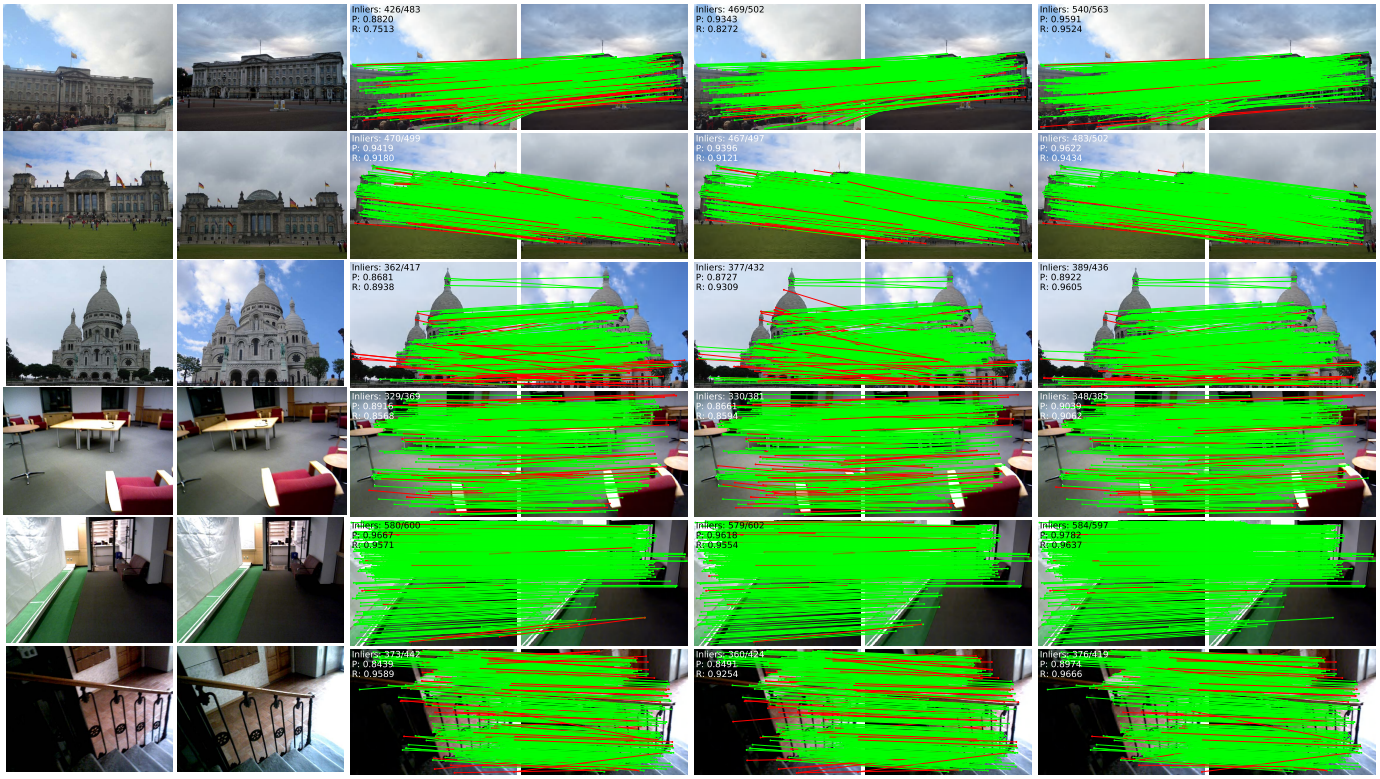
TABLE 1
Comparative results of **correspondence classification** task on the unknown YFCC100M and SUN3D datasets.

Datasets	YFCC100M			SUN3D		
Matcher	Precision (%)	Recall (%)	F-score	Precision (%)	Recall (%)	F-score
RANSAC [6]	43.55	50.65	46.83	44.87	48.82	46.76
MAGSAC [22]	45.15	62.36	50.26	44.41	54.46	50.01
LPM [25]	43.75	65.65	51.72	44.28	55.42	50.63
PointNet++ [29]	46.39	84.17	59.81	46.30	82.72	59.37
LFGC [13]	52.84	85.68	65.37	46.11	83.92	59.52
DFE [30]	54.00	85.56	66.21	46.18	84.01	59.60
ACNe [17]	55.62	85.47	67.39	46.16	84.01	59.58
OANet++ [16]	55.78	85.93	67.65	46.15	84.36	59.66
T-Net	58.21	86.38	69.55	47.27	84.16	60.54
T-Net++	59.16	86.35	70.21	47.81	84.27	61.01

Results on correspondence classification. Quantitative comparison results in terms of Precision, Recall, and F-score are shown in Table 1. We can see that the learning-based methods are able to achieve better performance on all evaluation metrics compared to the traditional methods, which demonstrates the effectiveness of the learning-based methods for dealing with large-scale datasets. Our T-Net and T-Net++ obtain better results compared with other methods, especially for T-Net++. Specifically, our T-Net++ is able to achieve 2.56% and 1.35% $F\text{-}score$ performance gains over the state-of-the-art OANet++ on both unknown outdoor and indoor scenes. Compared with T-Net, T-Net++ also shows better improvements on both datasets. In addition, we show some comparison visualization results on unknown outdoor and indoor scenes as shown in Fig. 5. Our proposed T-Net++ is able to get competitive correspondence classification results for several challenging scenes, such as large viewpoint and illumination changes, occlusions, blurs, and repetitive structures. These results well demonstrate the ability of our T-Net++ to find inliers from the putative feature point correspondence set containing a large number of outliers.

Results on camera pose estimation. Recovering accurate camera poses requires sufficient and appropriate inliers, which is essential for numerous feature matching based tasks. Thus, it is often used as the main metric for evaluating the performance of feature matching methods. We use off-the-shelf SIFT [3] feature extraction method to establish putative feature point correspondence set via the nearest-neighbor matching of feature descriptor. Quantitative comparison results with existing correspondence pruning works on both outdoor and indoor scenes are reported in Table 2.

We can see that our proposed methods are able to yield the best results under almost all settings. Compared with OANet++, our T-Net++ obtains 12.20% and 2.44% performance improvements on both unknown outdoor and indoor scenes without RANSAC, respectively. Compared with our T-Net, our T-Net++ also achieves better performance gains, especially for unknown scenes without RANSAC as post-processing. For using RANSAC as a post-processing step, our T-Net and T-Net++ still work well. SuperGlue [34] utilizes graph neural network and takes the entire matching image pair as inputs, thus, it needs more computational and storage overhead. Although SuperGlue can get more accurate camera poses compared to other methods, our method still outperforms it under almost all settings at lower computational cost. In addition, we observe that RANSAC as post-processing may harm



(a) Original images (b) OANet++ [16] (c) T-Net (d) T-Net++

Fig. 5. Visualization results of (a) original images, (b) OANet++ [16], (c) T-Net and (d) T-Net++. The top three examples come from the unknown YFCC100M dataset and rest three examples come from the unknown SUN3D dataset. The green lines represent recognized correct correspondences of network, while the red lines represent recognized false correspondences of network. The precision and recall are reported at the top left corner of each image pair.

TABLE 2
Comparative results of camera pose estimation on both known and unknown YFCC100M and SUN3D datasets. Results **without/with** RANSAC as a post-processing step under the error threshold of 5° ($i.e.$, mAP5°) are reported. The size of network models is shown in the second column.

Matcher	Datasets	YFCC100M (%)				SUN3D (%)			
	Size (MB)	Known Scene		Unknown Scene		Known Scene		Unknown Scene	
		-	RANSAC	-	RANSAC	-	RANSAC	-	RANSAC
RANSAC [6]	-	-	31.26	-	40.82	-	20.28	-	11.63
MAGSAC [22]	-	-	32.80	-	41.61	-	20.35	-	16.24
PointNet++ [29]	12	10.49	33.78	16.48	46.25	10.58	19.17	8.10	15.29
LFGC [13]	0.39	17.45	36.75	25.95	50.00	11.55	20.60	9.30	16.40
DFE [30]	0.40	19.13	36.46	30.27	51.16	14.05	21.32	12.06	16.26
ACNe [17]	0.41	29.17	40.32	33.06	50.89	18.86	22.12	14.12	16.99
OANet++ [16]	2.47	32.57	41.53	38.95	52.59	20.86	22.31	16.18	17.18
SuperGlue [34]	12.02	35.00	43.17	48.12	55.06	22.50	23.68	17.11	18.23
LMCNet [15]	0.93	33.73	40.39	47.50	55.03	19.92	21.79	16.82	17.38
MS^{2}DG-Net [20]	2.61	38.36	45.34	49.13	57.68	22.20	23.00	17.84	17.79
MSA-Net [35]	1.45	39.53	44.57	50.65	56.28	18.64	22.03	16.86	17.79
T-Net	3.78	42.99	45.25	48.20	55.85	22.38	22.96	17.24	17.57
T-Net++	4.11	42.69	45.91	51.15	55.95	23.48	23.76	18.62	17.70

TABLE 3
Performance comparison of our method with other baselines when using different feature extraction methods on both known and unknown YFCC100M datasets. Results **without/with** RANSAC post-processing under error thresholds of 5° and 20° ($i.e.$, mAP5° and mAP20°) are reported.

Features	Matcher	Known Scene				Unknown Scene			
		mAP5°		mAP20°		mAP5°		mAP20°	
		-	RANSAC	-	RANSAC	-	RANSAC	-	RANSAC
SuperPoint	LFGC [13]	12.18	30.25	34.75	52.13	24.25	42.57	52.70	66.89
	OANet++ [16]	29.52	35.72	53.76	57.75	35.27	45.45	66.81	70.99
	T-Net	34.21	36.60	57.30	59.10	40.08	47.83	70.42	72.46
	T-Net++	37.24	37.30	59.69	59.44	42.80	47.63	72.80	73.03
SIFT	LFGC [13]	17.45	36.75	39.75	58.91	25.95	50.00	55.46	70.20
	OANet++ [16]	32.57	41.53	56.89	63.91	38.95	52.59	66.85	72.99
	T-Net	42.99	45.25	66.33	67.35	48.20	55.85	74.18	75.73
	T-Net++	42.69	45.91	65.85	67.87	51.15	55.95	75.37	75.97

the performance of networks on the large-scale indoor scene. This is because the SUN3D dataset comes with severe lack of texture, repetitive patterns, and a large amount of self-similarity, which makes it very difficult for a feature extraction method to generate a reliable correspondence set. Therefore, RANSAC cannot further extract enough inliers for the essential matrix estimation.

In this paper, feature extraction is used for extracting keypoints and constructing corresponding descriptors, which is vital for establishing a reliable putative correspondence set and predicting accurate essential matrix. Therefore, we consider the influence of different feature extraction methods for networks on the camera pose estimation task. Here, we choose learning-based SuperPoint [5] as a comparison. SuperPoint utilizes a fully-convolutional model to train image keypoint detection and description, which is a self-supervised network for a large amount of multiple-view geometry problems in the field of computer vision. The comparison results between traditional SIFT and learning-based SuperPoint feature extraction methods on both known and unknown outdoor

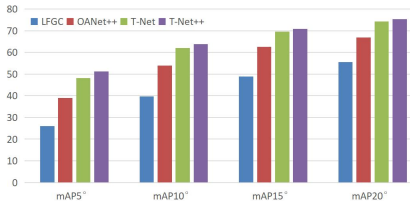


Fig. 6. Quantitative comparison of LFGC, OANet++, T-Net and T-Net++ for the mAP(%) with different error thresholds (*i.e.*, mAP5°, mAP10°, mAP15° and mAP20°) on the unknown outdoor scene.

scenes are given in Table 3. Learning-based LFGC and OANet++ are selected as baselines.

From the results in Table 3, we are surprised to find that SIFT works better than SuperPoint on all settings. We infer that SuperPoint is able to obtain better feature descriptors by using deep learning, but its keypoint localization is not accurate enough compared to SIFT. This conclusion is consistent with LFGC. However, our method still achieves the best performance under almost all settings for different feature extraction methods. Moreover, we show the comparison results of mAP with different error thresholds on the unknown outdoor scene in Fig. 6. It can be found that our proposed methods can obtain better results on different error thresholds.

TABLE 4
Quantitative comparison results for camera pose estimation with some contemporaneous methods on the YFCC100M dataset. mAP with different error thresholds is reported.

Methods	mAP5°	mAP10°	mAP20°
SuperGlue [34]	59.90	71.14	81.28
Patch2Pix [36]	47.85	57.95	68.59
PDC-Net [37]	63.98	73.48	81.91
SGMNet [38]	62.37	71.86	81.15
T-Net	64.90	75.09	84.01
T-Net++	65.75	75.73	84.39

Furthermore, we compare our work with some contemporaneous feature matching methods, including Patch2Pix [36], PDC-Net [37], and SGMNet [38], and report the comparison results in Table 4. Patch2Pix proposes a detect-to-refine manner to refine patch-level correspondence proposals for pixel-level correspondences. PDC-Net estimates dense pixel-level correspondences and reliability by a probabilistic model. SGMNet utilizes graph neural network to generate few reliable correspondences as seeds, and then propagates original keypoints to form dense correspondences. The first two methods operate directly on the matching image pairs and the last method focuses on matching keypoints and descriptors. For the comparative methods, we use publicly available network models to test their effectiveness. Our methods aim to identify inliers and outliers from initial correspondences, which can be embedded into many feature matching pipelines. Here, we adopt SuperGlue [34] to establish initial correspondences. We then utilize our methods to prune correspondences and MAGSAC++ [39] to recover camera poses. The model of Patch2Pix [36] is trained on the MegaDepth [40] dataset, therefore, its performance is not optimal. SuperGlue can provide better initial inputs for our methods. Therefore, T-Net and T-Net++ are able to achieve better performance compared to these competitors, demonstrating the effectiveness and flexibility of our methods.

TABLE 5
Ablation studies about the effectiveness of our proposed two key ideas on the unknown YFCC100M dataset. We report mAP5° (%) and mAP20° (%) **without/with** RANSAC as a post-processing step. **PointCN**: using the PointCN module. **CSSE**: using the CSSE module. **OA**: using the OA module. **T++**: using our T-Structure.

PointCN	CSSE	OA	T++	mAP5°	mAP20°
✓				25.95/50.00	55.46/70.20
	✓			31.43/50.03	58.73/70.23
✓		✓		33.08/51.18	60.72/72.09
	✓	✓		35.68/53.30	64.98/73.77
	✓	✓	✓	51.15/55.95	75.37/75.97

4.5 Ablation Studies

The core of our network is two key ideas: a novel architecture (T-Structure) adequately exploits all the feature information of each sub-network, and a CSSE module captures the potential relationship of channels and correspondences to boost the performance. In this section, we conduct ablation studies about our two key ideas and their details on the unknown outdoor scene for camera pose estimation task.

Two Key Ideas. To demonstrate the effectiveness of our proposed two key ideas, we verify the performance gains of our method on different combinations. As shown in Table 5, we select LFGC (PointCN) and OANet (PointCN + OA) as baselines. To demonstrate the effectiveness of the CSSE module, we only use the CSSE module to replace the PointCN module of baselines. The results show that our CSSE module can effectively improve the performance of baselines. To be specific, it achieves 5.48% improvement for LFGC and 2.60% improvement for OANet in terms of mAP5° without RANSAC as post-processing. When using our T-Structure, our T-Net++ (CSSE + OA + T++) achieves significant performance improvements. Specifically, it outperforms OANet by 18.07% and 14.65% in terms of mAP5° and mAP20° without RANSAC. This demonstrates the effectiveness of our two key ideas for two-view correspondence pruning.

TABLE 6
Ablation studies of three different iterative network structures on the unknown YFCC100M dataset. We report mAP5° (%) and mAP20° (%) **without/with** RANSAC as post-processing.

Method	mAP5°	mAP20°
OANet++	38.95/52.59	66.85/72.99
T-Net	48.20/55.85	74.18/74.88
T-Net++	51.15/55.95	75.37/75.97

In addition, we also report the comparison results of the three different iterative network structures in Table 6. OANet++ only adds the final outputs of the previous sub-network as additional inputs of the following sub-network. T-Net directly adopts a concatenation operation to integrate the feature information of all sub-networks. Our T-Net++ designs a novel Local-Global Attention Fusion (LGAF) module to better fuse the feature information of all sub-networks based on T-Net. We can see that T-Net++ achieves the best performance compared to the other two methods on both mAP5° and mAP20° without/with RANSAC as post-processing, which further verifies the effectiveness of our proposed method.

T-Structure. In our T-Net++, we design an improved network structure, named T-Structure, to largely improve network

TABLE 7
Ablation studies about different feature extraction manners of T-Structure on the unknown YFCC100M dataset. We report mAP5°(%) and mAP20° (%) **without/with** RANSAC as post-processing.

Method	mAP5°	mAP20°	Size(MB)
summation	50.63/55.88	75.01/75.72	3.85
concatenation	51.15/55.95	75.37/75.97	4.11

performances. For exploiting the feature information of all sub-networks, we first use a concatenation operation along the channel dimension of features to extract all information, and then adopt a novel LGAF module to integrate all features adequately. In addition to the concatenation operation, element-wise summation [41], [42] is also a common operation for information fusion in deep learning. Therefore, we replace the concatenation operation in T-Structure with the element-wise summation operation. The comparison results are shown in Table 7. Although the element-wise summation operation has a smaller network size compared to the concatenation operation, its performance is slightly dropped on unknown outdoor scenes. This might show that the concatenation operation is suitable for our T-Structure. So we use the concatenation operation to extract feature information.

TABLE 8
Ablation studies about the LGAF module on the unknown YFCC100M dataset. We report mAP5° (%) and mAP20° (%) **without/with** RANSAC as a post-processing step. LC: using the local context branch. GC_{avg}: using the global average context branch. GC_{max}: using the global maximal context branch. α_1: using α_1 in Eq. (9).

LC	GL_{avg}	GL_{max}	α_1	mAP5°	mAP20°
✓				49.95/55.35	74.52/75.19
	✓			48.28/55.55	73.67/75.70
		✓		48.58/55.00	74.43/75.47
	✓	✓		49.90/54.23	74.53/75.19
	✓	✓	✓	51.03/55.60	75.32/75.53
✓	✓	✓	✓	51.15/55.95	75.37/75.97

In our LGAF module, we utilize a three-branch structure to capture the local and global context of the concatenation feature to enhance its representation ability. The LGAF module contains three main branches and a learned parameter: the local context (LC) branch, the global average context (GC_{avg}) branch, the global maximal context (GC_{max}) and α_1 in Eq. (9). Their performance gains are reported in Table 8. We can see that, the performance of network will slightly degrade when the three branches are used individually. Moreover, the learned parameter α_1 is valid for balancing two types of global contexts. Our complete LGAF (LC + GC_{avg} + GC_{max} + α_1) module can get the best performance in terms of mAP5° and mAP20°, which proves the rationality of three main branches and a learned parameter.

Our T-Structure is able to integrate valuable information of each sub-network for correspondence pruning. We show the output results of each sub-network and the final output results of our network in Fig. 7, to further reveal the effectiveness of our T-Structure. We can see that each sub-network has a different focus, and our T-Net++ is able to integrate the focus of these sub-networks to get better results, which demonstrates the rationality and validity of our T-Net++.

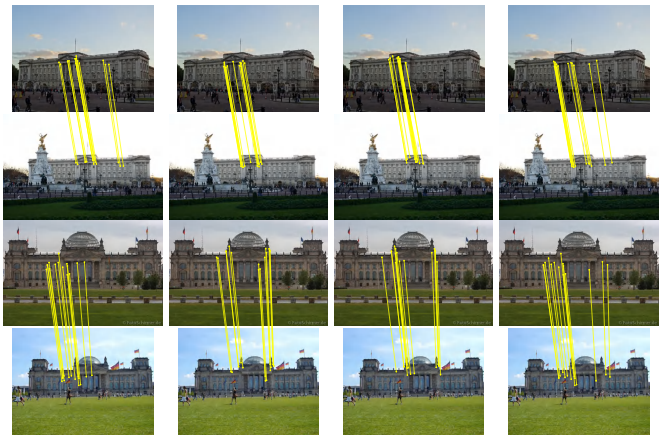


Fig. 7. Visual results of top 15 response correspondences in the same image pair from different unknown outdoor testing scenes. From left to right: the results of the first sub-network, the second sub-network, the third sub-network, and the final output. We draw the correspondences in yellow if they conform to the ground truth epipolar geometry.

TABLE 9
Ablation studies about the CSSE layer on the unknown YFCC100M dataset. We report mAP5° (%) and mAP20° (%) **without/with** RANSAC as a post-processing step. Rc_{avg}: using the average channel relationship operation. Rc_{max}: using the maximum channel relationship operation. α_2: using α_2 in Eq. (17). Rs: using the spatial relationship operation.

Rc_{avg}	Rc_{max}	α_2	Rs	mAP5°	mAP20°
✓				48.60/54.63	73.85/75.00
	✓			48.88/55.50	74.16/75.88
✓	✓			49.58/55.40	74.82/75.29
✓	✓	✓		50.00/55.78	74.82/75.69
✓	✓	✓	✓	51.15/55.95	75.37/75.97

CSSE Module. As mentioned in Section 3.3, we develop a novel CSSE module to replace the PointCN block used in previous methods for better network learning. In the CSSE module, a CSSE layer is used for capturing the potential relationship of channels and correspondences, which consists of three main operations and a learned parameter: the average channel relationship (Rc_{avg}) operation, the maximum channel relationship (Rc_{max}) operation, the spatial relationship (Rs) operation and α_2 in Eq. (17). Their performance gains are reported in Table 9. We can find that they have similar performance when we only use the average or maximum channel relationship operations. When we combine them, the performance will be improved. Meanwhile, the learned parameter α_2 also improves the effectiveness of network. Our complete CSSE (Rc_{avg} + Rc_{max} + Rs + α_2) layer can obtain the best performance on the unknown YFCC100M dataset, which proves the rationality of three main operations and a learned parameter.

Inlier ratio. The inlier ratio represents the proportion of inliers within the initial correspondences, which can significantly impact the performance of RANSAC [6] and its variants. Hence, we evaluate the influence of the inlier ratio for our Tnet++. We first employ SIFT to detect 2000 keypoints and construct corresponding descriptors. Then, Lowe's ratio test [3] with different thresholds is adopted to control the inlier ratio of initial correspondences. Finally, we test the performance of Tnet++ on the

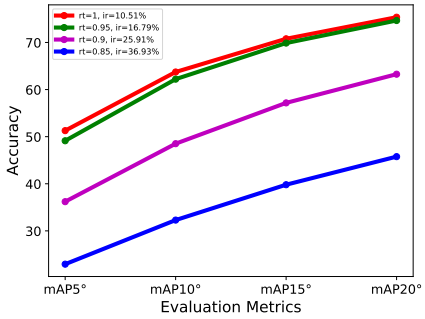


Fig. 8. The influence of different inlier ratios (ir) controlled by the ratio test (rt) with different thresholds on the unknown YFCC100M dataset. mAP under error thresholds of $5°$, $10°$, $15°$ and $20°$ are reported.

unknown YFCC100M dataset. As illustrated in Fig. 8, our method performs best in the case of lower inlier ratios. While the ratio test is beneficial for reducing outliers in initial correspondences, it also leads to the removal of many important inliers. The results indicate that our network has the ability to retain sufficient inliers in the presence of a large number of outliers for estimating a reliable essential matrix, demonstrating the robustness of method. This is an advantage of the learning-based methods over the traditional hypothesize-and-verify approaches.

4.6 Extended Tasks

In this section, for measuring the generalization ability, we extend the correspondence pruning methods to two feature matching based tasks, $i.e.$, remote sensing image registration and point cloud registration.

4.6.1 Remote Sensing Image Registration

The goal of image registration is to maximize the alignment of the overlapping region between the reference image and the sensed image. It is a fundamental but challenging problem in the remote sensing community, which also needs accurate correspondences to determine a reliable geometric transformation. That is, correspondence pruning can provide accurate candidate correspondences for transformation estimation. Therefore, we test the performance of methods for the remote sensing image registration task.

Datasets. We choose 55 pairs of low-altitude remote sensing images acquired from the unmanned aerial vehicle with different imaging types and target scenes from [43], [44]. These matching image pairs usually face severe viewpoint changes and extreme patterns. Similar to the outdoor and indoor images, SIFT is used to construct the putative correspondences of these image pairs. The inlier labels as landmarks to evaluate the accuracy of image registration are manually annotated, where the average number of putative correspondences and average inlier ratio are $1,214.5$ and 32.53%, respectively.

Evaluation. Because of the lack of enough remote sensing images for training, we directly use the network models trained on the YFCC100M dataset to test their generalizations. We use correspondence pruning networks to process the initial correspondences, and RANSAC with 50 iterations is equipped as a post-processing step to estimate the geometric transformation. We adopt $root\ mean\ square\ error\ (RMSE)$, $maximum\ error\ (MAE)$ and $median\ error\ (MEE)$ to measure the performance of methods with the following definitions:

$$RMSE = \sqrt{1/L \sum_{i=1}^{L} \|r_i^c - \mathcal{F}(s_i^c)\|_2^2}, \tag{30}$$

$$MAE = max\left\{\|r_i^c - \mathcal{F}(s_i^c)\|_2\right\}_{i=1}^{L}, \tag{31}$$

$$MEE = median\left\{\|r_i^c - \mathcal{F}(s_i^c)\|_2\right\}_{i=1}^{L}, \tag{32}$$

where r_i^c and s_i^c represent the corresponding landmarks ($i.e.$, pixel coordinates) of the reference image and the sensed image, respectively. $\mathcal{F}$ denotes the transformation function from the sensed image to the reference image, which is estimated using extracted correspondences of corresponding methods. In this paper, we select Thin Plate Spline [45] as the transformation parameterizing due to its generality and smooth functional mapping nature. L is the number of selected landmarks. $\|\cdot\|_2$ denotes the Euclidean norm of vector. max and $median$ are functions of getting the maximal and median values of a set, respectively.

TABLE 10
Registration results on remote sensing data. The average $RMSE$, MAE, MEE and runtime (RT) are used for evaluation. ↓ means that a lower value is better.

Method	$RMSE$↓	MAE↓	MEE↓	$RT(ms)$↓
RANSACN-1k [6]	80.21	228.56	102.70	371.15
RANSACN-2k	55.74	158.47	69.48	735.11
LFGC [13]	30.72	81.78	39.67	52.36
OANet++ [16]	14.13	54.02	16.12	61.01
SuperGlue [34]	19.86	66.98	23.95	102.09
MS^{2}DG-Net [20]	9.54	45.36	12.53	61.53
MSA-Net [35]	7.56	38.55	8.43	62.08
T-Net	8.48	41.70	9.36	62.27
T-Net++	4.17	32.12	3.18	61.59

Results. Here, we select RANSAC with 1k and 2k iterations, LFGC, OANet++, SuperGlue, MS^{2}DG-Net and MSA-Net as comparative methods. We first report the visualization results of correspondence classification and image registration on some representative image pairs in Fig. 9. From these visual results, we can see that our proposed T-Net++ can detect more accurate inliers and obtain more satisfying registration performance than our T-Net for challenging remote sensing scenes.

Table 10 outlines the average values in terms of $RMSE$, MAE, MEE and runtime (RT) for image registration on 55 selected image pairs. From the quantitative results, it is shown that our proposed T-Net++ can obtain the best $RMSE$, MAE and MEE, while maintaining a decent RT. For RANSAC, it may be more likely to fail when the putative correspondences contain a large number of outliers. In fact, RANSAC can theoretically obtain better performance when increasing the number of iterations, but this would require more runtime. Moreover, SuperGlue shows worse generalization ability due to the use of feature descriptors, which easily leads to failed results in some scenarios excluding its training sets. In contrast, our T-Net++ achieves good generalization ability due to our well-designed network structure.

4.6.2 Point Cloud Registration

The goal of point cloud registration is to estimate the transformation between two given point clouds with the same or similar scene, which is a key step for point cloud processing. Similar to image feature matching, the point cloud registration task can

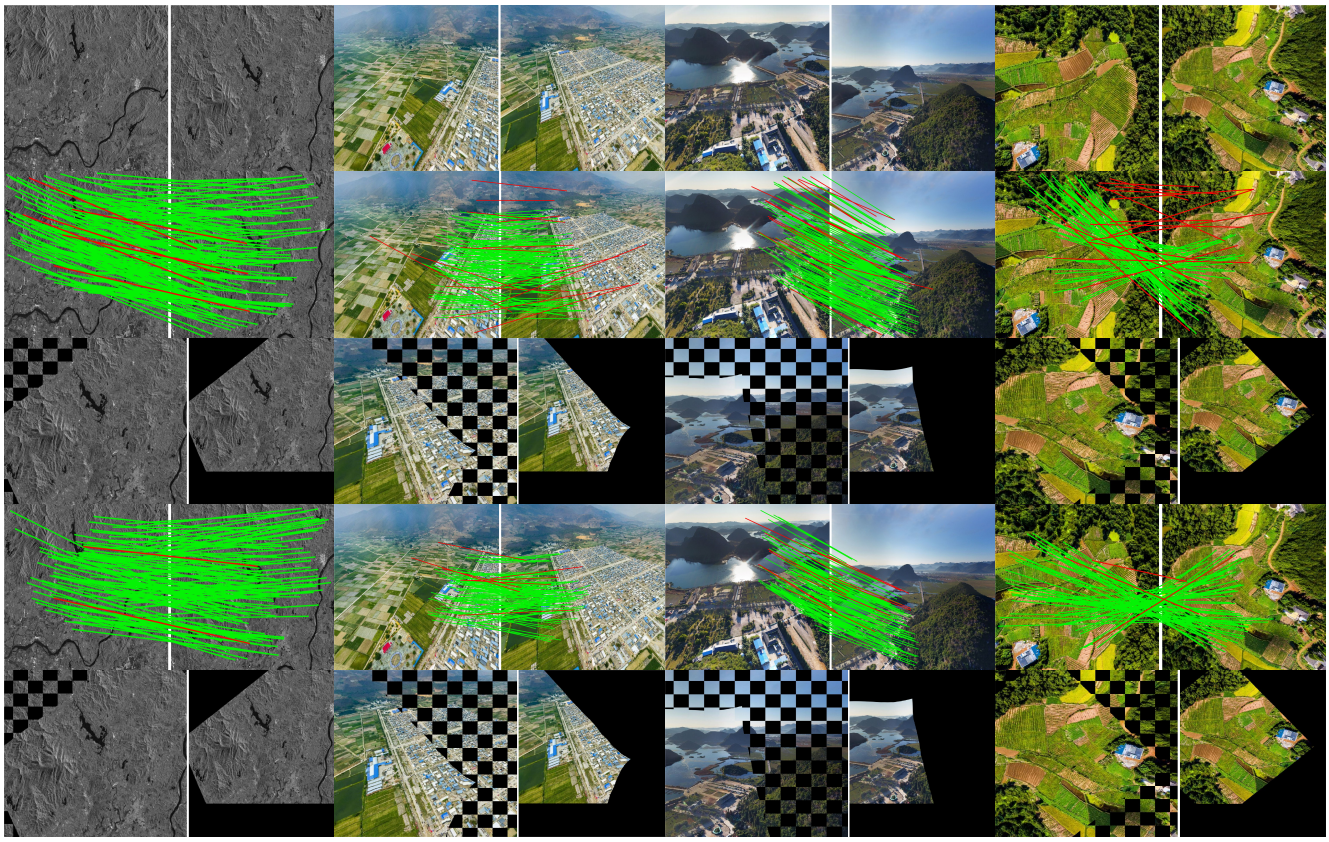


Fig. 9. Correspondence classification and image registration results on 4 representative image pairs. From top to bottom: The first row presents the original input images, where the left and right in each group are reference and sensed images. The 2nd and 4th rows are correspondence classification results of our proposed T-Net and T-Net++, respectively. Green line: true-positive; red line: false-positive. For visibility, at most 200 randomly selected correspondences are shown. Their corresponding image registration results are shown in the 3rd and 5th rows, where the left and right in each group are checkerboard results and the warped sensed images, respectively.

be solved by establishing a reliable point cloud correspondence set, in which correspondence pruning is one of the indispensable steps for the successful feature-based point cloud registration task. Therefore, we extend the $2D$ correspondence pruning method to $3D$ point clouds.

Datasets. In this paper, we select the 3DMatch [46] dataset (indoor settings) with different feature descriptors. The testing set includes $1,623$ partially overlapped point cloud fragments on eight scenes. Due to the different dimensions of correspondences, we need to retrain the correspondence pruning networks.

Evaluation. Following the setting of PointDSC [47], we apply our correspondence pruning network to replace the classification module in PointDSC. Learned descriptor FCGF [48] and traditional descriptor FPFH [49] are adopted to construct putative point cloud correspondences. Network models are retrained on FCGF. Each point cloud pair establishes 1000 correspondences for training and testing. The batchsize is set to 16 with 50 epochs. Adam optimizer is employed to optimize the network, initializing it with a learning rate of 10^{-4} and applying an exponential decay factor of 0.99. For evaluating the performance., following [50], $registration\ recall\ (RR)$, $rotation\ error\ (RE)$ and $translation\ error\ (TE)$ are used for evaluating the accuracy of methods with the following definitions:

$$RE(\hat{\mathbf{R}}) = arccos\frac{\mathrm{Tr}\left(\hat{\mathbf{R}}^{\mathbf{T}}\mathbf{R}^{*}\right)-1}{2}, \tag{33}$$

$$TE(\hat{\mathbf{t}}) = \left\|\hat{\mathbf{t}}-\mathbf{t}^{*}\right\|_{2}, \tag{34}$$

where $\hat{\mathbf{R}}$ and $\hat{\mathbf{t}}$ are the predicted rotation and translation of methods, respectively. $\mathbf{R}^{*}$ and $\mathbf{t}^{*}$ represent the ground-truth rotation and translation, respectively. RR indicates the registration accuracy of RE less than $15°$ and TE less than 30cm. We also use $Precision$ and $Recall$ to evaluate the performance of methods for the point cloud correspondence pruning.

TABLE 11
Comparative results of point cloud registration on 3DMatch [46]. The average RR, RE, TE, $Precision$ and $Recall$ are used for evaluation.

Descriptor	FCGF (learned)					FPFH (traditional)				
Matcher	RR (%)	RE (°↓)	TE (cm↓)	Precision (%)	Recall (%)	RR (%)	RE (°↓)	TE (cm↓)	Precision (%)	Recall (%)
RANSAC [6]	86.57	3.16	9.67	76.86	77.45	40.05	5.16	13.65	51.52	34.31
DGR [50]	91.30	2.40	7.48	67.47	78.94	69.13	3.78	10.80	28.80	12.42
LFGC [13]	91.56	2.04	6.47	77.49	80.85	73.66	2.13	6.58	64.60	58.67
OANet++ [16]	91.87	2.02	6.51	77.76	80.49	72.51	2.10	6.49	62.62	55.96
MS^{2}DG-Net [20]	92.05	2.04	6.50	77.76	84.35	78.54	2.13	6.72	67.99	72.10
MSA-Net [35]	91.90	2.05	6.53	77.38	81.99	72.49	2.12	6.77	64.02	59.33
T-Net	91.93	2.05	6.45	77.99	82.23	72.96	2.09	6.41	63.34	56.59
T-Net++	92.42	2.02	6.45	78.39	82.51	77.14	2.09	6.55	68.77	67.06

Results. We report the comparative results on two settings in Table 11. From the quantitative results, we can find that our proposed T-Net++ can obtain decent performance in all settings. The improvement of T-Net++ on the traditional descriptor FPFH is significant compared to some networks. To be specific, T-Net++ outperforms T-Net by 4.18% on RR. Moreover, we show some visualization results for correspondence pruning and registration in Figs. 10 and 11, respectively, which further demonstrates the effectiveness of our proposed methods. The quantitative and visual

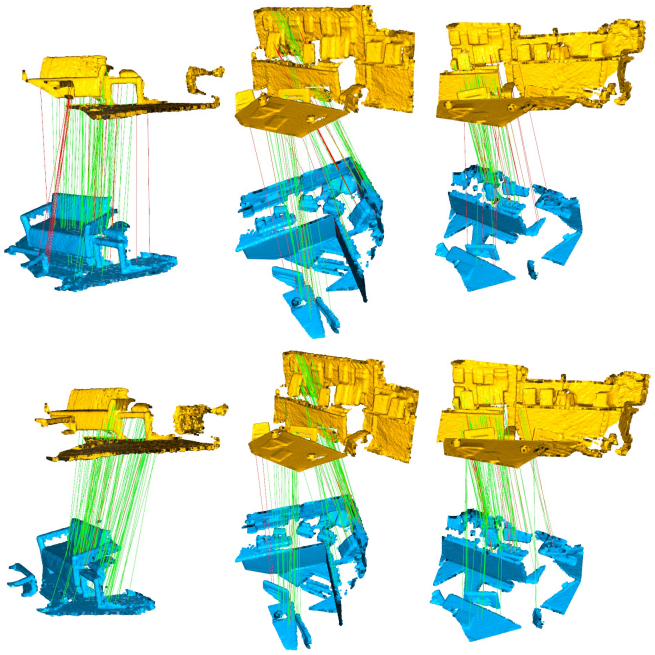


Fig. 10. Visualization of correspondence pruning resluts on the 3DMatch dataset with different scenes. The first and second rows are results of T-Net and T-Net++, respectively. The green lines represent recognized inliers, while the red lines represent recognized outliers by networks.

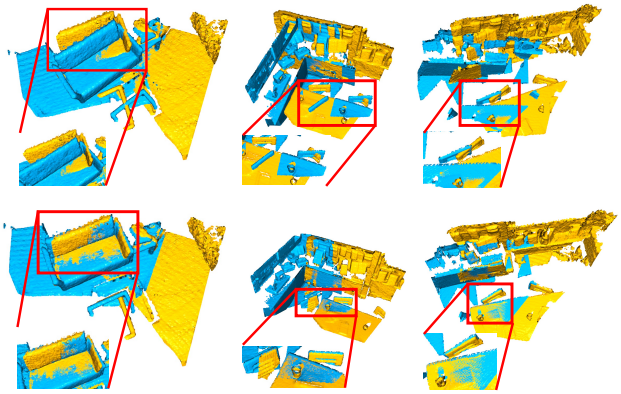


Fig. 11. Corresponding visualization results of point cloud registration on the 3DMatch dataset. The first and second rows are results of T-Net and T-Net++, respectively.

comparative results well show the generalization ability of our method.

5 Conclusion

In this work, we develop a simple yet effective permutation-equivariance framework, T-Net++, for the task of two-view correspondence pruning. T-Net++, consisting of a " – " structure and a "|" structure, adeptly integrates valuable feature information from iterative sub-networks. To enhance its capabilities, we introduce a new LGAF module designed to maximize the utilization of concatenated features. Additionally, we propose a novel CSSE module, which serves as the backbone of our network. This module captures potential relationship cues from both the channel and spatial dimensions of features, ultimately elevating the representation of crucial features. These operations empower our network to gather rich contextual information, enabling the establishment of reliable correspondences and relative poses. Our experimental results across various computer vision tasks demonstrate that T-Net++ outperforms state-of-the-art methods, achieving remarkable performance improvements. However, it is worth noting that the T-Structure network can inevitably lead to increased computational overhead when using too many iterative sub-networks. In future work, we aim to develop a more efficient network structure for correspondence pruning without sacrificing accuracy. Furthermore, we intend to explore the inherent properties of correspondence, such as motion coherence, for distinguishing inliers and outliers.

References

[1] 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.

[2] Y. Wang, C. Yan, Y. Feng, S. Du, Q. Dai, and Y. Gao, "Storm: Structure-based overlap matching for partial point cloud registration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 1135 – 1149, 2023.

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

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

[5] 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.

[6] 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.

[7] D. Barath, D. Rozumnyi, I. Eichhardt, L. Hajder, and J. Matas, "Finding geometric models by clustering in the consensus space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5414–5424.

[8] Z. Li, J. Ma, and G. Xiao, "Density-guided incremental dominant instance exploration for two-view geometric model fitting," *IEEE Transactions on Image Processing*, vol. 32, pp. 5408–5422, 2023.

[9] H. Wang, G. Xiao, Y. Yan, and D. Suter, "Searching for representative modes on hypergraphs for robust geometric model fitting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 3, pp. 697–711, 2019.

[10] 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.

[11] A. Fan, X. Jiang, Y. Ma, X. Mei, and J. Ma, "Smoothness-driven consensus based on compact representation for robust feature matching," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 8, pp. 4460–4472, 2023.

[12] G. Xiao, S. Wang, H. Wang, and J. Ma, "Mining consistent correspondences using co-occurrence statistics," *Pattern Recognition*, vol. 119, p. 108062, 2021.

[13] 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.

[14] 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 Conference on Computer Vision and Pattern Recognition*, 2019, pp. 215–224.

[15] 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.

[16] 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 International Conference on Computer Vision*, 2019, pp. 5845–5854.

[17] 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.

[18] J. Chen, S. Chen, X. Chen, Y. Dai, and Y. Yang, "Csr-net: Learning adaptive context structure representation for robust feature correspondence," *IEEE Transactions on Image Processing*, vol. 31, pp. 3197–3210, 2022.

[19] Z. Zhong, G. Xiao, S. Wang, L. Wei, and X. Zhang, "Pesa-net: Permutation-equivariant split attention network for correspondence learning," *Information Fusion*, vol. 77, pp. 81–89, 2022.

[20] L. Dai, Y. Liu, J. Ma, T. Lai, C. Yang, and R. Chen, "Ms2dg-net: Progressive correspondence learning via multiple sparse semantics dynamic graph," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

[21] 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 International Conference on Computer Vision*, 2021, pp. 1950–1959.

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

[23] D. Barath and J. Matas, "Graph-cut ransac: Local optimization on spatially coherent structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 4961 – 4974, 2022.

[24] J. Ma, J. Zhao, J. Tian, A. L. Yuille, and Z. Tu, "Robust point matching via vector field consensus," *IEEE Transactions on Image Processing*, vol. 23, no. 4, pp. 1706–1721, 2014.

[25] 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.

[26] X. Jiang, J. Ma, J. Jiang, and X. Guo, "Robust feature matching using spatial clustering with heavy outliers," *IEEE Transactions on Image Processing*, vol. 29, pp. 736–746, 2019.

[27] 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.

[28] 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.

[29] 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.

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

[31] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, no. 5828, pp. 133–135, 1981.

[32] 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.

[33] 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.

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

[35] 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.

[36] 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.

[37] 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.

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

[39] 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.

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

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

[42] 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.

[43] 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.

[44] 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.

[45] G. Wahba, *Spline models for observational data*. SIAM, 1990.

[46] 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.

[47] 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.

[48] 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.

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

[50] C. Choy, W. Dong, and V. Koltun, "Deep global registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2514–2523.

Guobao Xiao received the B.S. degree in information and computing science from Fujian Normal University, China, in 2013 and the Ph.D. degree in Computer Science and Technology from Xiamen University, China, in 2016. From 2016-2018, he was a Postdoctoral Fellow in the School of Aerospace Engineering at Xiamen University, China. He is currently a Professor at Hangzhou Dianzi University, China. He has published over 50 papers in journals and conferences including IEEE TPAMI/TIP, IJCV, ICCV, ECCV, etc. His research interests include machine learning, computer vision and pattern recognition. He has been awarded the best PhD thesis award in China Society of Image and Graphics (a total of ten winners in China). He also served on the program committee (PC) of CVPR, ICCV, ECCV, etc.

Xin Liu received the masters 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. He is also a visiting student at the Hangzhou Dianzi University. His research interests include computer vision and image matching.

Zhen Zhong received the bachelor's degree in traditional Chinese medicine from the Hunan University of Chinese Medicine, Hunan, China, in 2018, and the M.S. degree from Fujian University of Traditional Chinese Medicine, Fuzhou, China, in 2022. He is a visiting student at Hangzhou Dianzi University. His research interests include computer vision, machine learning, and pattern recognition.

Xiaoqin Zhang received the B.Sc. degree in electronic information science and technology from Central South University, China, in 2005 and Ph.D. degree in pattern recognition and intelligent system from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, China, in 2010. He is currently a professor in Wenzhou University, China. His research interests are in pattern recognition, computer vision and machine learning. He has published more than 100 papers in international and national journals, and international conferences, including IEEE T-PAMI, IJCV, IEEE T-IP, IEEE T-NNLS, IEEE T-C, ICCV, CVPR, NIPS, IJCAI, AAAI, ACM MM and among others.

Jiayi Ma (M'14-SM'21) received the B.S. degree in information and computing science and the Ph.D. degree in control science and engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2008 and 2014, respectively. He is currently a Professor with the Electronic Information School, Wuhan University. He has authored or co-authored more than 200 refereed journal and conference papers, including IEEE TPAMI/TIP, IJCV, CVPR, ICCV, ECCV, *etc*. His research interests include computer vision, machine learning, and pattern recognition. Dr. Ma has been identified in the 2019-2021 Highly Cited Researcher lists from the Web of Science Group. He is an Area Editor of *Information Fusion*, an Associate Editor of *Neurocomputing*, *Sensors* and *Entropy*.

Haibin Ling received B.S. and M.S. from Peking University in 1997 and 2000, respectively, and Ph.D. from University of Maryland in 2006. From 2000 to 2001, he was an assistant researcher at Microsoft Research Asia; from 2006 to 2007, he worked as a postdoctoral scientist at UCLA; from 2007-2008, he worked for Siemens Corporate Research as a research scientist; and from 2008 to 2019, he was a faculty member of the Department of Computer Sciences for Temple University. In fall 2019, he joined the Department of Computer Science of Stony Brook University, where he is now a SUNY Empire Innovation Professor. His research interests include computer vision, augmented reality, medical image analysis, visual privacy protection, and human computer interaction. He received Best Student Paper Award of ACM UIST in 2003 and NSF CAREER Award in 2014. He serves as associate editors for IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), Pattern Recognition (PR), and Computer Vision and Image Understanding (CVIU). He has served as Area Chairs various times for CVPR and ECCV.